

普通高等教育“十二五”规划教材

电气工程·自动化专业规划教材

# 欧姆龙 CP1 系列 PLC 原理与应用

王冬青 主编

丁 锋 主审

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书以欧姆龙(OMRON)CP1 系列可编程序控制器为背景,系统阐述了 CP1 系列 PLC 的硬件组成、工作原理和指令系统,包括中断类指令和脉冲输出指令,还介绍了任务编程方式、模拟量输入/输出单元的使用、CX-P 编程软件的使用和功能块编程。详细讲解了 PLC 控制系统的设计方法,并列出了常用的编程实例。本书还特别介绍了欧姆龙小型多功能变频器 3G3MX2 的使用方法。本书内容新颖,语言通俗易懂,理论联系实际。为了便于教学与自学,配有一定数量的习题,并编写了内容丰富的实验指导。本书还配有丰富的 PPT 课件。

本书可作为高等院校自动化、电气技术、机电一体化及相关专业的教材,也可以作为工程技术人员继续教育的参考书或 PLC 的培训教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容  
版权所有·侵权必究

## 图书在版编目(CIP)数据

欧姆龙 CP1 系列 PLC 原理与应用 / 王冬青主编. —北京: 电子工业出版社, 2011.10  
电气工程·自动化专业规划教材  
ISBN 978-7-121-14584-1

I. ①欧… II. ①王… III. ①可编程序控制器—程序设计—高等学校—教材 IV. ①TP332.3

中国版本图书馆 CIP 数据核字(2011)第 185777 号

策划编辑: 史鹏举

责任编辑: 史鹏举 文字编辑: 王 纲

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 21.25 字数: 614 千字

印 次: 2011 年 10 月第 1 次印刷

定 价: 38.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010)88254888。

质量投诉请发邮件至 [zltts@phei.com.cn](mailto:zltts@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010)88258888。

# 前 言

可编程控制器(Programmable Logic Controller, PLC)是集计算机技术、自动控制技术、通信技术为一体的自动控制装置。PLC 作为工业自动化的三大支柱(PLC、工业机器人、CAD/CAM)之一,已被广泛地应用于工业控制的各个领域。

目前,PLC 产品大致可分成美国、欧洲和日本三个流派。在我国 PLC 市场上,日本的小型机产品销量较大。CP1(CP1H/CP1L/CP1E)系列 PLC 是日本欧姆龙公司自 2005 年陆续推出的整体式小型机,其中 CP1H 是凝聚了多种功能的小型高功能 PLC,其 CPU 单元具有 20/40 点型,是一款性价比高、功能完备的通用控制器。其特点是结构紧凑,集成了开关量控制、模拟量控制、高速计数、高速脉冲输出、串行通信等功能于一身。CP1L/CP1E 具有经济、简单和高效的特点,有丰富的 CPU 单元,具有 10/14/20/30/40/60 点型。

CP1 系列 PLC 指令丰富,采用任务编程方式(CP1E 除外)并支持功能块编程方法使编程更加清楚、简便。CP1 系列 PLC 取消了手持编程器配置,没有通常的外设接口,取而代之的是 USB 端口。最多有两个串行通信功能端口,可自由选择 RS232C 和 RS485。

本书以 CP1H PLC 为典型代表,阐述了 CP1 系列 PLC 的硬件组成、工作原理和指令系统,详细介绍了中断类指令和脉冲输出指令的应用,还介绍了任务编程方式、模拟量输入/输出单元的使用、CX-P 编程软件的使用和功能块编程。书中讲解了 PLC 控制系统的设计方法,列举了常用的编程实例。本书还特别介绍了欧姆龙小型多功能变频器 3G3MX2 的使用方法。

目前欧姆龙公司主推的小、中和大型机分别为:CP1(CP1H/CP1L/CP1E)、CJ1 和 CS1 系列。CP1 系列 PLC 的内部存储器设置及指令系统与 CJ1/CS1 系列 PLC 几乎相同。所不同的是,CP1 系列是整体结构,而 CJ1 和 CS1 是模块式结构。第 2 章介绍了 CJ1/CS1 的 I/O 通道分配,所以本书介绍的编程和应用也适用于 CJ1 和 CS1 系列 PLC。

本书内容新颖,语言通俗易懂,理论联系实际。为便于教学与自学,配有一定数量的习题,并编写了内容丰富的实验指导。

本书配有 PPT 等教学资源,可登录电子工业出版社华信教育资源网 [www.hxedu.com.cn](http://www.hxedu.com.cn),免费注册下载。

本书编写过程中,欧姆龙自动化(中国)有限公司提供了设备和大力支持,在此致以衷心感谢。

全书共 9 章,由王冬青执笔,韩卓玉参与了第 8 章的编写。本书由丁锋教授审阅并定稿。参加本书编写工作的还有初燕云、李健、王丽美、高聪聪、王杰、孟敬、孙永康。

由于编者水平有限,错误和疏漏之处在所难免,敬请读者批评指正。

E-mail: dqwang64@163.com

编 者

# 目 录

第 1 章 可编程控制器概述 . . . . .	(1)	3.3 顺序输入/输出指令及指令的形式 . . . . .	(43)
1.1 PLC 的基本概念 . . . . .	(1)	3.3.1 顺序输入/输出指令 . . . . .	(43)
1.1.1 PLC 的定义 . . . . .	(1)	3.3.2 指令的微分/非微分形式和即时刷新形式 . . . . .	(48)
1.1.2 PLC 的特点 . . . . .	(2)	3.4 时序控制指令 . . . . .	(48)
1.2 PLC 的基本组成 . . . . .	(3)	3.4.1 结束指令 END 和空指令 NOP . . . . .	(48)
1.3 PLC 的编程语言 . . . . .	(8)	3.4.2 互锁 IL/互锁解除 ILC 指令 . . . . .	(49)
1.4 PLC 的工作方式和性能指标 . . . . .	(11)	3.4.3 暂存继电器 TR . . . . .	(51)
1.4.1 PLC 的循环扫描工作方式 . . . . .	(11)	3.4.4 跳转 JMP/跳转结束 JME 指令 . . . . .	(52)
1.4.2 PLC 的主要性能指标 . . . . .	(13)	3.4.5 其他顺序控制指令 . . . . .	(53)
1.5 PLC 产品及资料下载 . . . . .	(14)	3.5 定时器/计数器指令 . . . . .	(54)
习题 . . . . .	(15)	3.5.1 定时器 TIM/TIMH 指令 . . . . .	(54)
第 2 章 CP1 系列 PLC 的组成 . . . . .	(16)	3.5.2 计数器 CNT/CNTR 指令 . . . . .	(55)
2.1 CP1 系列 PLC 的基本构成 . . . . .	(16)	3.6 数据传送指令 . . . . .	(58)
2.1.1 CP1 系列 PLC 的主机 . . . . .	(16)	3.6.1 传送 MOV/@MOV、求反传送 MVN/@MVN 指令 . . . . .	(60)
2.1.2 CP1 系列 PLC 的其他功能 . . . . .	(21)	3.6.2 块设置指令 BSET . . . . .	(60)
2.2 CP1 系列 PLC 的扩展单元 . . . . .	(24)	3.7 数据比较指令 . . . . .	(61)
2.2.1 连接 CP1W 和 CPM1A 系列扩展单元 . . . . .	(24)	3.7.1 常用比较指令 . . . . .	(61)
2.2.2 连接 CJ1 系列高性能单元 . . . . .	(26)	3.7.2 条件类比较指令 . . . . .	(63)
2.3 CP1 系列 PLC 的存储器分配 . . . . .	(27)	3.7.3 时刻类比较指令 . . . . .	(64)
2.3.1 CP1H/CP1L 系列 PLC 的 I/O 存储器分配 . . . . .	(27)	3.7.4 其他比较指令 . . . . .	(65)
2.3.2 CP1E 系列 PLC 的 I/O 存储器分配 . . . . .	(31)	3.8 数据移位指令 . . . . .	(66)
2.4 CJ1/CS1 的 I/O 通道分配 . . . . .	(32)	3.8.1 数据移位寄存器指令 SFT . . . . .	(68)
2.4.1 CJ1 的 I/O 通道分配 . . . . .	(32)	3.8.2 可逆移位寄存器指令 SFTR/@SFTR . . . . .	(69)
2.4.2 CS1 的 I/O 通道分配 . . . . .	(34)	3.8.3 数字左、右移位指令 SLD/@SLD、SRD/@SRD . . . . .	(70)
习题 . . . . .	(35)	3.8.4 字移位指令 WSFT/@WSFT . . . . .	(71)
第 3 章 CP1 系列 PLC 的指令系统 . . . . .	(37)	3.9 数据转换指令 . . . . .	(71)
3.1 指令概述 . . . . .	(37)	3.9.1 BCD 码→BIN 码转换指令 BIN/@BIN . . . . .	(73)
3.2 基本指令及编程 . . . . .	(39)		
3.2.1 基本指令 . . . . .	(39)		
3.2.2 基本编程规则和编程方法 . . . . .	(41)		



3.9.2	BIN 码→BCD 码转换指令 BCD/@BCD	(74)	4.2.2	CX-P 工程	(124)
3.9.3	4→16/8→256 译码指令 MLPX	(74)	4.2.3	CX-P 视图	(128)
3.9.4	16→4/256→8 编码指令 DMPX	(75)	4.2.4	其他常用的操作	(134)
3.9.5	ASCII 码转换指令 ASC	(76)	4.3	CX-P 编程	(135)
3.9.6	七段译码指令 SDEC	(76)	4.3.1	建立一个新工程	(135)
3.10	递增/递减指令	(77)	4.3.2	生成符号和地址	(135)
3.11	数据运算指令	(78)	4.3.3	程序编辑	(136)
3.11.1	十进制运算指令	(78)	4.3.4	程序编译	(140)
3.11.2	二进制运算指令	(83)	4.3.5	在线工作	(140)
3.12	逻辑运算指令	(85)	4.4	CX-P 的功能块编程	(143)
3.13	浮点转换、运算指令	(87)	4.4.1	创建功能块	(143)
3.13.1	浮点数	(87)	4.4.2	功能块的使用	(152)
3.13.2	浮点数与二进制数间的转换 指令	(90)	习题		(158)
3.13.3	浮点数加+F、减-F、乘*F、除/F 运算指令	(92)	第 5 章	任务编程方法	(159)
3.13.4	角度↔弧度变换指令	(96)	5.1	任务程序概述	(159)
3.13.5	三角函数运算指令	(98)	5.2	任务的使用方法	(162)
3.13.6	平方根 SQRT、指数 EXP、对数 LOG 指令	(101)	5.3	中断任务	(166)
3.13.7	浮点数运算的应用举例	(103)	5.3.1	CP1 系列 PLC 的中断功能	(166)
3.14	子程序控制指令	(104)	5.3.2	输入中断	(170)
3.14.1	子程序调用 SBS、子程序开始 SBN、 子程序返回 RET 指令	(105)	5.3.3	高速计数器中断	(173)
3.14.2	宏指令 MCRO	(107)	5.3.4	脉冲输出控制指令	(186)
3.15	步进控制指令	(108)	5.3.5	间隔定时器中断	(195)
3.15.1	步启动 SNXT 指令、步定义 STEP 指令	(108)	5.3.6	综合例子	(196)
3.15.2	步进程序的结构及程序的编写 规则	(109)	习题		(200)
3.15.3	步进程序的执行过程	(111)	第 6 章	PLC 控制系统的设计	(201)
3.15.4	步进控制实例	(113)	6.1	概述	(201)
习题		(117)	6.1.1	PLC 控制系统设计的基本 步骤	(201)
第 4 章	编程软件 CX-P	(121)	6.1.2	PLC 的应用程序	(202)
4.1	CX-P 简介	(121)	6.2	程序设计方法	(203)
4.2	CX-P 的使用	(122)	6.2.1	逻辑设计法	(203)
4.2.1	CX-P 主窗口	(122)	6.2.2	时序图设计法	(205)
			6.2.3	顺序控制设计法	(208)
			6.2.4	经验设计法	(218)
			6.2.5	继电器控制电路图转换 设计法	(222)
			6.2.6	具有多种工作方式系统的编程 方法	(224)
			6.3	应用程序实例	(231)
			6.3.1	自动定时搅拌系统	(231)

6.3.2	定位及减速控制	• • • •	(233)	8.2.2	3G3MX2 系列变频器	• • • • •	273
6.3.3	汽车自动清洗机	• • • •	(233)	8.3	3G3MX2 变频器的运行控制	• • • • •	280
6.3.4	霓虹灯控制电路	• • • •	(234)	8.3.1	变频器操作面板的操作	• • • • •	280
6.3.5	三层载货电梯控制	• • • •	(236)	8.3.2	3G3MX2 运行的操作面板	• • • • •	282
6.3.6	自动售货机的控制	• • • •	(239)	8.3.3	3G3MX2 运行的外部电路	• • • • •	283
6.3.7	送料小车的控制	• • • •	(242)	8.3.4	CP1H 通过输出端子直接控制	• • • • •	290
6.3.8	传送带控制系统	• • • •	(243)	8.3.5	CP1H 通过 Modbus-RTU 简易主站	• • • • •	296
6.3.9	配料槽控制系统	• • • •	(244)	8.3.6	利用 CX-Drive 软件控制	• • • • •	301
6.3.10	U 形板折板机的 PLC	• • • • •	(246)	变频器	• • • • •	• • • • •	304
6.3.11	并行序列顺序控制	• • • •	(250)	习题	• • • • •	• • • • •	304
习题	• • • • •	• • • • •	(252)	第 9 章	PLC 实验	• • • • •	307
第 7 章	CP1 系列 PLC 的模拟量功能	• • • • •	255	9.1	熟悉 PLC 及 CX-P 编程软件	• • • • •	307
7.1	CP1 系列 PLC 的内置模拟量输入/输出	• • • • •	255	9.2	定时器/计数器指令	• • • • •	311
7.1.1	模拟量输入/输出功能介绍	• • • • •	255	9.3	三相异步电动机的 PLC 控制	• • • • •	313
7.1.2	模拟量输入/输出功能的使用	• • • • •	258	9.4	互锁/跳转/数据比较/数据移位/数据	• • • • •	314
7.2	CP1 系列 PLC 的扩展模拟量输入/输出	• • • • •	260	传送指令	• • • • •	• • • • •	317
7.2.1	CP1W 和 CPM1A 系列模拟量	• • • • •	260	9.5	彩灯的 PLC 控制	• • • • •	318
7.2.2	CPM1A -MAD01 模拟量 I/O	• • • • •	261	9.6	数据转换及数据运算指令	• • • • •	322
习题	• • • • •	• • • • •	266	9.7	十字路口交通灯控制	• • • • •	324
第 8 章	欧姆龙 3G3MX2 变频器	• • • • •	267	9.8	子程序指令与中断控制指令	• • • • •	326
8.1	变频器概述	• • • • •	267	9.9	三人抢答器控制程序	• • • • •	328
8.1.1	变频器的构成	• • • • •	267	9.10	变频器的基本运行	• • • • •	329
8.1.2	变频器的分类	• • • • •	269	9.11	变频器的多段速运行	• • • • •	332
8.1.3	变频器的控制方式	• • • • •	270	参考文献	• • • • •	• • • • •	(332)
8.1.4	变频器的主要功能	• • • • •	272				
8.2	欧姆龙 3G3MX2 变频器	• • • • •	272				
8.2.1	欧姆龙变频器简介	• • • • •	272				

# 第 1 章 可编程控制器概述

## 1.1 PLC 的基本概念

### 1.1.1 PLC 的定义

可编程控制器(Programmable Controller)是为工业控制应用而设计制造的。早期的可编程控制器称为可编程逻辑控制器(Programmable Logic Controller, PLC),如今人们仍然沿用 PLC 作为可编程控制器的缩写。它是一个以微处理器为核心的数字运算操作的电子系统装置,专为工业现场应用而设计,它采用可编程的存储器,用于在其内部存储执行逻辑运算、顺序控制、定时/计数和算术运算等操作指令,并通过数字式或模拟式的输入、输出接口,控制各种类型的机械或生产过程。

1987 年 2 月国际电工委员会(International Electrotechnical Commission, IEC)发布了可编程控制器标准草案第三稿,其中对 PLC 定义如下:可编程控制器是一种数字运算操作电子系统,专为在工业环境下应用而设计。它采用了可编程序的存储器,用来在其内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令,并通过数字的、模拟的输入和输出,控制各种类型的机械或生产过程。可编程序控制器及其有关的外围设备,都应按易于与工业控制系统形成一个整体、易于扩充其功能的原则设计。

20 世纪 60 年代末,由于市场的需要,工业生产从大批量、少品种的生产方式转变为小批量、多品种的生产方式。但是,当时这种大规模生产线的控制大多是继电器控制系统,体积大,耗电多,容易损坏,而且改变生产程序非常困难。为了改变这种状况,满足生产产品多样性的要求,1968 年美国通用汽车公司公开对外招标,要求用新的电气控制装置取代继电器控制装置,以适应灵活、快速地改变生产程序的要求。

1969 年美国数字设备公司(Digital Equipment Corporation, DEC)研制成功了世界上第一台 PLC,并在美国通用汽车的生产线上试验成功,从此开辟了可编程逻辑控制器的时代。1971 年,日本引进了这项技术,很快就研制出了日本第一台 PLC。1973 年,原西德和法国也研制出了自己的 PLC。20 世纪 70 年代中期,欧美及日本的 PLC 产品中多以微处理器及大规模集成电路芯片为其核心部件,使 PLC 的功能进一步扩展,并且有了自诊断功能,可靠性得到了进一步提高。随着微电子技术的迅猛发展,20 世纪 80 年代中期,PLC 的处理速度和可靠性大大提高,不仅增加了多种特殊功能,而且体积进一步缩小,成本大幅度下降。到 20 世纪 90 年代中期之后,PLC 几乎完全计算机化,其速度更快,功能更强,PLC 的各种智能化模块不断被开发出来,一些厂家还推出了 PLC 的计算机辅助编程软件,许多小型 PLC 的性能也不可小视。

现在,PLC 不仅能进行逻辑控制,在模拟量的闭环控制、数字量的智能控制、数据采集、监控、通信连网及集散控制等方面都得到了广泛的应用。如今大、中型,甚至小型 PLC 都配有 A/D、D/A 转换及算术运算功能,有的还具有 PID 功能。这些功能使 PLC 应用于模拟量的闭环控制、运动控制、速度控制等具有了硬件基础;PLC 具有输出和接收高速脉冲的功能,配合相应的传感器及伺服装置,PLC 可以实现数字量的智能控制;PLC 配合可编程终端设备(如触摸屏),可以实时显示采集到的现场数据及分析结果,为分析和研究系统提供依据;利用 PLC 的自检信号可实现系统监控;PLC 具有较强的通信功能,可与计算机或其他智能装置进行通信和连网,从而能方便地实现集散控制。功能完备的 PLC 不仅能满足控制的要求,而且能满足现代化大生产管理的需要。

目前,世界上一些著名电器生产厂家几乎都在生产 PLC,产品功能日趋完善,换代周期越来越短。为了进一步扩大 PLC 在工业自动化领域的应用范围,适应大中小型企业不同需要,PLC 产品大致向两个方向发展:小型 PLC 向体积缩小、功能增强、速度加快、价格低廉的方向发展,使之能更加广泛地取代继电器控制,更便于实现机电一体化;大中型 PLC 向高可靠性、高速度、多功能、网络化的方向发展,将 PLC 系统的控制功能和信息管理功能融为一体,使之能对大规模、复杂系统进行综合性的自动控制。

### 1.1.2 PLC 的特点

PLC 作为一种专为工业环境应用而设计的数字运算操作电子系统,其优越性主要体现在以下几个方面。

#### 1. 控制功能强

PLC 所采用的 CPU 一般是具有较强位处理功能的位处理机,为了增强其复杂的控制功能和通信联网等管理功能,可采用双 CPU 的运行方式,使其功能得到极大的加强。

#### 2. 可靠性高,抗干扰能力强

继电器控制系统中,由于器件的老化,脱焊、触点的抖动及触点电弧等现象是不可避免的,大大降低了系统的可靠性。继电器控制系统的维修工作不仅耗资费时,而且由于停产维修所造成的损失也不可估量。而在 PLC 控制系统中,大量的开关动作是由无触点的半导体电路来完成的,加之 PLC 在硬件和软件方面都采取了强有力的措施,使产品具有极高的可靠性和抗干扰能力,故此 PLC 可以直接安装在工业现场而稳定地工作。从国内外使用 PLC 的实际情况来看,平均无故障率可以达到几万甚至几十万小时以上。因而 PLC 被誉为“专为适应恶劣的工业环境而设计的计算机”。

#### 3. 编程语言简单易学

PLC 采用的是梯形图编程语言,与早期的继电器控制图非常相似,理解方式相同,非常清晰直观,了解继电器控制技术的电气人员很容易学会。

#### 4. 适用于恶劣的工业环境

PLC 采用封装的方式,适合于各种振动、腐蚀、污染、有毒气体的应用场合。

#### 5. PLC 与外部设备连接方便

PLC 上采用统一接线方式的可拆装的端子排,不同端子适合不同的功能。例如,PLC 的输入接口可以直接与各种输入设备(如按钮、各种传感器等)连接,输出接口具有较强的驱动能力,可以直接与继电器、接触器、电磁阀等强电电器连接,接线简单,使用非常方便。

#### 6. PLC 的扩展能力强

通过增加扩展机架使 PLC 可用于控制从几个 I/O 点到几百个 I/O 点的控制系统。同时也可以根据工业现场的需要,给 PLC 增加一些特殊功能的模块,如以太网模块、模拟量模块、位置模块、运动控制模块等。

#### 7. 体积小、质量轻、易于实现机电一体化

由于 PLC 内部电路主要采用半导体集成电路,所以具有结构紧凑、体积小、质量轻、功耗低的特点。更由于其能适应各种恶劣的环境和具备很强的抗干扰能力,因而成为实现机电一体化十分理想的控制装置。

## 8. 性价比高

与其他的控制方式相比, PLC 控制系统设计周期短, 可以在实验室进行模拟调试, 使现场的工作量大大减少, 省时省力。

## 1.2 PLC 的基本组成

PLC 是以微处理器为核心的一种特殊的工业用计算机, 其结构与一般的计算机相类似, 由中央处理单元(CPU)、存储器(RAM、ROM、EPROM、EEPROM 等)、输入接口、输出接口、I/O 扩展接口、外部设备接口及电源等组成。

根据结构形式不同, PLC 可分为整体式(也称箱体式)和组合式(也称模块式)两类。整体式 PLC 是将中央处理单元(CPU)、存储器、输入单元、输出单元、电源、通信接口、I/O 扩展接口等组装在一个箱体内构成主机。另外还有独立的扩展单元等与主机配合使用。整体式 PLC 的结构紧凑、体积小, 小型机常采用这种结构, 基本组成如图 1.1 所示。日本欧姆龙(OMRON)公司的 CPM1A、CPM2A 和 CP1 系列(CP1 H/CP1L/CP1E)均为整体式 PLC。

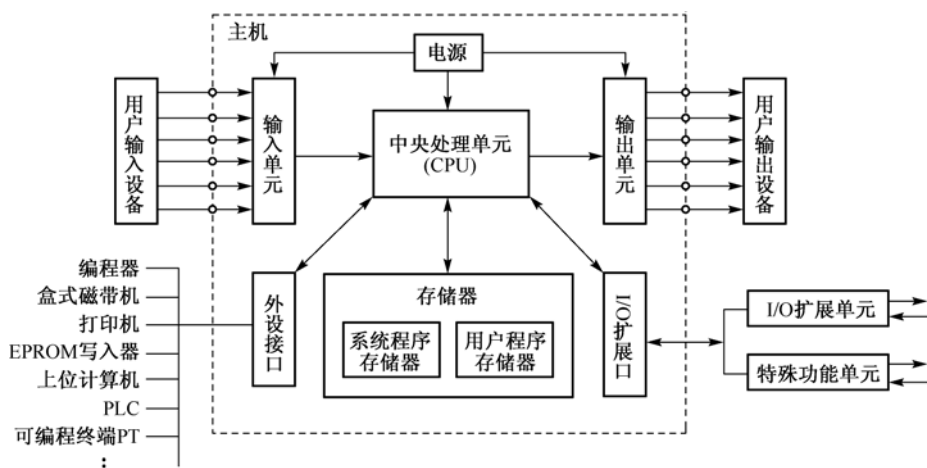


图 1.1 整体式 PLC 的组成示意图

组合式 PLC 的组成如图 1.2 所示。这种结构的 PLC 是将 CPU 单元、输入单元、输出单元、智能 I/O 单元和通信单元等分别做成相应的电路板或模块, 各模块可以插在底板上, 模块之间通过底板上的总线相互联系。装有 CPU 的单元称为 CPU 模块, 其他单元称为扩展模块。CPU 与各扩展模块之间若通过电缆连接, 距离一般不超过 10 m。中、大型机常采用组合式。日本欧姆龙(OMRON)公司的 C200H、CJ1M 和 CS1 等为组合式 PLC。

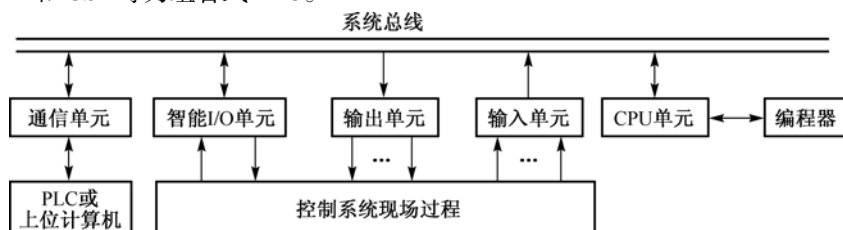


图 1.2 组合式 PLC 的组成示意图

本书将重点介绍的 CP1(包括 CP1H、CP1L 和 CP1E)系列 PLC 为整体式, CP1H 单元具有 20 点和 40 点规格的输入/输出电路, CP1L/CP1E CPU 单元具有 10 点、14 点、20 点、30 点、40 点和 60 点规

格的输入/输出电路。CP1 系列 PLC 将模块式的各个单元集成为一体,不如模块式灵活但是使用方便。如果 I/O 点数不够,可用 CP1 系列的基本 I/O 扩展单元 CP1W(或 CPM1A 和 CPM2A 的基本 I/O 扩展单元 CPM1A)进行扩展。不同型号的 CP1 系列 PLC 可以连接的扩展单元的类型和数量不同。另外,CP1 系列 PLC 可以连接 CJ 系列特殊 I/O 单元或 CPU 总线单元。

下面以 CP1H 为例,介绍 CP1 系列 PLC 各组成部分及其作用。PLC 的具体型号不同时,其组成部分及作用会稍有差异。CP1 系列不同 PLC 的差异将在第 2 章中介绍。

### 1. 中央处理单元(CPU)

CPU 是 PLC 的核心部件,它类似人的大脑,能指挥 PLC 按照预先编好的系统程序完成各种任务,其作用有以下几点。

- ① 接收、存储由编程工具输入的用户程序和数据,并可通过显示器显示出程序的内容和存储地址。
- ② 检查、校验用户程序。对正在输入的用户程序进行检查,发现语法错误立即报警,并停止输入;在程序运行过程中若发现错误,则立即报警或停止程序的执行。
- ③ 接收、调用现场信息。将接收到现场输入的数据保存起来,在需要该数据的时候将其调出并送到需要该数据的地方。
- ④ 执行用户程序。当 PLC 进入运行状态后,CPU 根据用户程序存放的先后顺序,逐条读取、解释和执行程序,完成用户程序中规定的各种操作,并将程序执行的结果送至输出端,以驱动 PLC 外部的负载。
- ⑤ 故障诊断。诊断电源、PLC 内部电路的故障,根据故障或错误的类型,通过显示器显示出相应的信息,以提示用户及时排除故障或纠正错误。

### 2. 存储器

存储器可以分为以下两种。

#### (1) 系统程序存储器

系统程序是厂家根据其选用 CPU 的指令系统编写的,它决定了 PLC 的功能。系统程序一般由厂家直接固化在只读 ROM 或 EPROM 之中,用户不能更改其内容。它包括如下内容。

检查程序: PLC 加电后,由检查程序检查 PLC 各部件操作是否正常,并显示。

翻译程序:将用户输入的程序变换成由微机指令组成的程序,然后再执行。

监控程序:相当于总控程序,根据用户的需要调用相应的内部程序。

#### (2) 用户程序存储器

用户根据控制要求而编制的应用程序称为用户程序。不同机型的 PLC,其用户程序存储器的容量可能差异较大。用户程序经常需要改动,所以用户程序存储器必须可读/写。用户程序和数据存储在随机存取存储器 RAM 中。一般要用后备电池(锂电池)进行掉电保护,以防掉电时丢失程序和数据。CP1H 系列整体式 PLC,可选外置快闪存储器卡,容量达 512 KB,并含有内置闪存,用户程序和参数区(如 PLC 设置)是自动备份和自动恢复的。

### 3. 输入/输出单元

输入/输出(I/O)单元(或 I/O 模块)是 PLC 与外部设备相互联系的窗口。输入单元接收现场设备向 PLC 提供的信号,例如,由按钮、操作开关、限位开关、继电器触点、接近开关、拨码器等提供的开关量信号,可使用开关量输入单元接收这些开关量信号;可使用模拟量(A/D)输入单元接收温度传感器及变送器发出的直流电压信号;可使用高速计数单元接收旋转编码器发出的脉冲信号等。输入单元将接收到的各种现场信号转换成 CPU 能够接收和处理的信号。输出单元通常有开关量输出、脉冲输出和模拟量输出 3 种输出单元。开关量输出单元用于驱动控制继电器、接触器、电磁阀及指示灯等。

脉冲输出单元用于连接步进电动机驱动器，驱动步进电动机，构成定位控制系统；也可以连接具有脉冲输入端口的交流伺服驱动器，驱动交流永磁同步电动机，构成位置闭环控制系统。模拟量(D/A)输出单元可用于连接具有模拟量输入端口的直流驱动器，驱动直流电动机构成直流调速系统或位置控制系统；也可用于连接具有模拟量输入端口的交流伺服驱动器，驱动交流伺服电动机，构成速度控制系统、位置控制系统或同步控制系统等。

(1) 数字量(开关量)输入单元

欧姆龙公司的 CP1H 带有内置输入/输出端子。X/XA 型 CPU 单元的数字量(开关量)输入单元的接点占用输入继电器 0CH 的 00~11 位 12 点，和 1CH 的 00~11 位 12 点，共计 24 点。因为 0CH/1CH 的高位 12~15 通常被系统清除，故不可作为内部辅助继电器使用。输入电路如图 1.3 所示，性能指标见表 1.1。图 1.3(a)中点画线框内为输入电路，框外左侧为现场用户接线；3.0 kΩ 为限流电阻；910 Ω 与 1000 pF 构成滤波器，用以除去输入信号中的高频干扰；虚线框内为光电耦合器输入指示灯，由发光二极管和光敏三极管组成，它将输入电路与内部电路(控制电路)隔离，提高输入单元的抗干扰能力；输入指示灯在外部输入电路接通时亮，表示有信号输入。图 1.3(b)、图 1.3(c)原理与图 1.3(a)一样，并且可以看出直流电源的极性接法是任意的。输入可以是按钮开关、行程开关及位置开关等，使用时外接 24 V 直流电源，如图 1.3 所示，也可以是光敏类接近开关和磁敏类接近开关等开关元件。

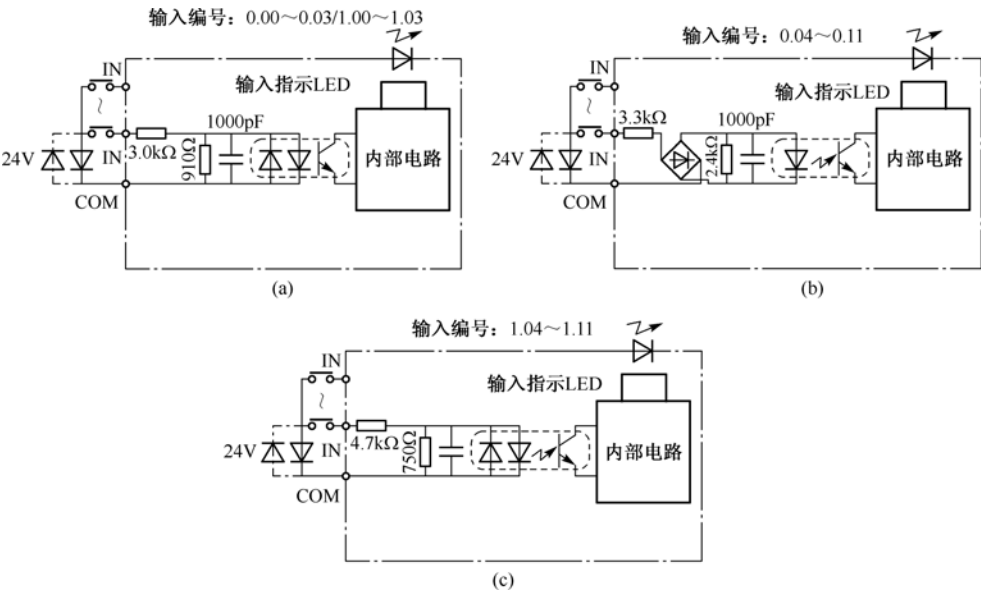


图 1.3 CP1H-XA 型数字量(开关量)输入单元电路图

表 1.1 CP1H-XA 型数字量(开关量)输入单元性能指标

项 目	规 格		
	0.04~0.11	0.00~0.03/1.00~1.03	1.04~1.11
输入电压	DC 24 V、+10%、-15%		
对象传感器	2 线式		
输入阻抗	3.3 kΩ	3.0 kΩ	4.7 kΩ
输入电流	7.5 mA TYP	8.5 mA TYP	5 mA TYP
ON 电压	最小 DC 17.0 V 以上	最小 DC 17.0 V 以上	最小 DC 14.4 V 以上
OFF 电压/电流	最大 DC 5.0 V 1 mA 以下	最大 DC 5.0 V 1 mA 以下	最大 DC 5.0 V 1 mA 以下
ON 响应时间	2.5 μs 以下	50 μs 以下	1 ms 以下
OFF 响应时间	2.5 μs 以下	50 μs 以下	1 ms 以下

(2) 数字量(开关量)输出单元

开关量输出单元有继电器输出单元、晶体管输出单元及双向晶闸管(可控硅)输出单元。

① 继电器输出单元。X/XA 型 CPU 单元的数字量(开关量)输出继电器占用 100CH 的 00~07 位 8 点, 101CH 的 00~07 位 8 点, 共计 16 点。100CH/101CH 的高位位 08~15 可作为内部辅助继电器使用。图 1.4 是电路图。内部有 16 只小型 24 V 直流继电器, 其规格性能指标见表 1.2。图中点画线框内是继电器单元的输出电路, 框外右侧为现场用户接线, 图中仅画出 1 位接点的输出电路, 其他各继电器输出点的输出电路均相同。

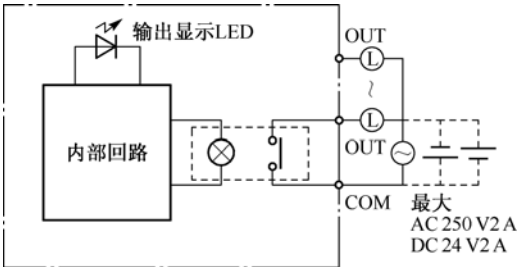


图 1.4 CP1H-XA 型数字量(开关量)输出单元电路图

表 1.2 CP1H-XA 型继电器输出单元性能指标

项    目			规    格
最大开关能力			AC 250 V/2 A ( $\cos\phi=1$ ) DC 24 V/2 A (4 A/公共端)
最小开关能力			DC 5V、10 mA
继电器寿命	电    气	阻性负载	10 万次 (DC 24 V)
		感性负载	48000 次 (AC 240 V $\cos\phi=0.4$ )
	机    械		2000 万次
ON 响应时间			15 ms 以下
OFF 响应时间			15 ms 以下

继电器输出单元的负载可以是接触器、牵引电磁铁、比例电磁铁、气动电磁阀、液压电磁阀、信号灯及警报器等。外接电源视负载类型而定, 可选用直流或交流电源。使用直流电源时, 电源极性接法任意。

② 晶体管输出单元。CP1H-XA 的晶体管输出单元电路如图 1.5 所示, 性能指标见表 1.3。

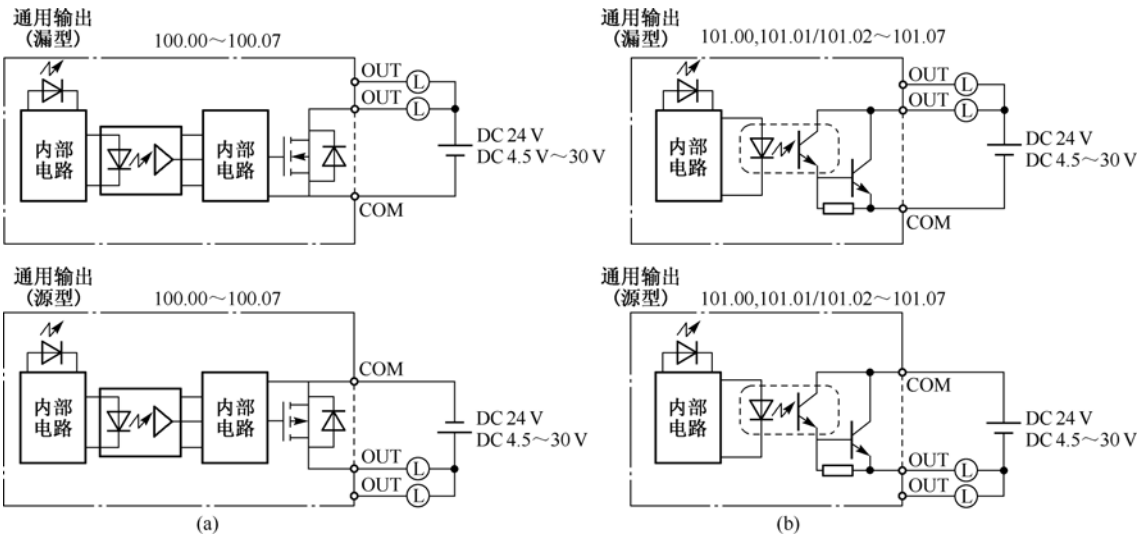


图 1.5 CP1H-XA 型晶体管输出单元电路图



表 1.3 CP1H-XA 型的晶体管输出单元性能指标

项 目	规 格		
	100.00～100.07	101.00, 101.01	101.02～101.07
最大开关能力	DC 4.5～30 V、300 mA/点、0.9 A/公共、3.6 A/单元		
最小开关能力	DC 4.5～30 V 1 mA		
漏电流	0.1 mA 以下		
残留电压	0.6 V 以下	1.5 V 以下	
ON 响应时间	0.1 ms 以下		
OFF 响应时间	0.1 ms 以下		1 ms 以下
熔断丝	有(1 个/点)		

图1.5(a)、图1.5(b)分别对应地址 100CH 的 8 个和 101CH 的 8 个输出端子的输出电路，图1.5中点画线框内是晶体管输出单元内部的输出电路，框外右侧为现场用户接线。外接电源为 24 V 直流电源。图 1.5(b)虚线框内为光电耦合器，外面的三极管为功率无触点开关元件，用于接通或断开负载电路。图中只画出一位即一个输出点的输出电路，其他各个输出点的输出电路均相同。

晶体管作为无触点开关元件，寿命长且响应时间短，如本例中晶体管输出单元 ON 响应时间为 0.1 ms 左右，OFF 响应时间为 0.1 ms 左右。最大通断范围为 DC 1~300 mA，4.5~30 V。通常晶体管输出单元采用光电耦合器作为输出级，使内部电路(控制电路)与输出电路隔离，内部电路不直接受到负载电流的影响，提高了输出单元抗干扰能力。对于晶体管输出单元，当负载为感性负载时，应采用浪涌吸收器，用于吸收浪涌电压。此外，还应有过电流和过载保护电路。

4. 电源部分

PLC 中一般配有开关式稳压电源为内部电路供电。开关电源的输入电压范围宽、体积小、质量轻、效率高、抗干扰性能好。有的 PLC 能向外部提供 24 V 的直流电源，可给输入单元所连接的外部开关或传感器供电。

5. 外设端口

每台 PLC 都有外设端口。通过外设端口，PLC 可与外部设备相连接。例如，连接编程器以输入、修改用户程序或监控程序的运行；有的 PLC 可以通过外设端口与其他 PLC、计算机或终端设备 PT 等连接进行通信，或连成各种网络等。

CP1 系列之前的 OMRON PLC 的外设端口，可直接连接手持编程器，向 PLC 写入程序。CP1 系列 PLC 取消了手持编程器配置，没有通常的外设端口，有一个外围设备 USB 端口。另外，还可以安装相应的选件板得到 RS-232C 和 RS-422A/485 两个通信端口。

6. 编程工具

编程工具是开发应用和检查维护 PLC 及监控系统运行不可缺少的外部设备。编程工具的主要作用是用来编辑程序、调试程序和监控程序的执行，还可以在线测试 PLC 的内部状态和参数，与 PLC 进行人机对话等。通常使用的编程工具可以是简易手持编程器，也可以是配有专用编程软件包的通用计算机。

(1) 简易编程器

简易编程器不能直接输入梯形图程序，只能输入语句表程序。用简易编程器编程时，编程器必须与 PLC 相连接。简易编程器的优点是价格低、体积小、质量轻、方便携带。

编程器可以不参与现场运行，所以一台编程器可以供多台 PLC 使用。

## (2) 计算机辅助编程

目前各 PLC 厂家都设计了 PLC 的计算机辅助编程软件。当 PLC 与装有编程软件的计算机连接通信时，可进行计算机辅助编程(见第 4 章)。编程软件的功能很强，它可以编辑、修改用户的程序，监控系统运行，打印文件，采集和分析数据，监控系统运行状况，对工业现场和系统进行仿真，将程序储存在磁盘上，实现计算机和 PLC 之间的程序相互传送等。

CP1 系列之前的 OMRON PLC 可直接采用手持编程器编程，也可以通过计算机辅助编程。CP1 系列 PLC 取消了手持编程器编程功能，只能通过 CX-Programmer(简称 CX-P)编程软件在计算机上编程，然后下载到 PLC 中。

## 7. I/O 扩展端口和扩展单元

当主机上的 I/O 点数或类型不能满足用户需要时，主机可以通过 I/O 扩展端口连接基本 I/O 扩展单元来增加 I/O 点。没有 I/O 扩展端口的 PLC 不能进行 I/O 点扩展。另外，通过 I/O 扩展端口还可以连接各种高功能 I/O 扩展单元和 CPU 高功能单元，扩展 PLC 的功能。

CP1 系列 PLC 的高功能 I/O 扩展单元和 CPU 单元有：A/D 单元、D/A 单元、高速计数单元、位置控制单元、温度控制单元和各种通信单元等。

# 1.3 PLC 的编程语言

可编程控制器是通过程序对系统进行控制的，各种机型的 PLC 都有自己的编程语言。PLC 编程语言有多种，其中 3 种是图形化语言，包括梯形图(Ladder Diagram, LD)、功能块(Function Block, FB)、顺序功能图(Sequential Function Chart, SFC)；两种是文本化语言，包括语句表(Instruction List, IL)和结构文本(Structured Text, ST)。

## 1. 梯形图 LD 编程语言

梯形图编程语言是一种图形语言，是若干图形符号的组合。不同厂家的 PLC 各有自己的一套梯形图符号。这种编程语言具有继电器控制电路形象、直观的优点，使熟悉继电器控制的技术人员很容易掌握。可见，各种机型的 PLC 都把梯形图作为第一编程语言。

表 1.4 列出了物理的继电器与 OMRON PLC 输出继电器的梯形图符号。







图 1.6(a)是用继电器控制的电动机直接启、停(也称启保停)控制梯形图，图 1.6(b)是用 PLC 控制的梯形图程序。由图可见，这两种梯形图形式很相似。但是，它们只是形式上的相似，实质上却存在着本质的差别，其主要区别有以下几点。

### (1) 两种继电器的区别

① 继电器控制电路中使用的继电器都是物理的电器，继电器与其他控制电器间的连接必须通过硬接线来完成；PLC 的继电器不是物理的电器，它是 PLC 内部的寄存器位，常称为“软继电器”。之所以称为“软继电器”，是因为它具有与物理继电器相似的功能。例如，当它的“线圈”通电时，其所属的常开触点闭合、常闭触点断开，当它的“线圈”断电时，其所属的常开触点和常闭触点均恢复常态。PLC 梯形图中的接线称为“软接线”，这种“软接线”是通过程序来实现的。

② PLC 的每一个继电器都对应着内部的一个寄存器位，由于可以无限次地读取某位寄存器的内容，所以，可以认为 PLC 的继电器有无数个常开、常闭触点可供用户使用。而物理继电器的触点个数是有限的。

表 1.4 两种继电器符号对照

类型		物理继电器	PLC 继电器
部件名称			
线圈			
触点	常开		
	常闭		

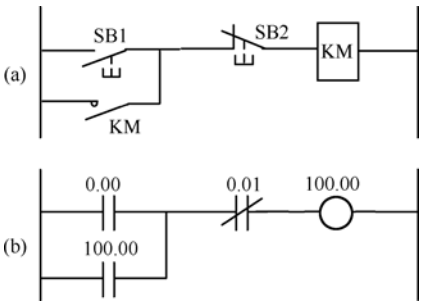


图 1.6 两种控制方式的梯形图

③ PLC 的输入继电器是由外部信号所驱动的，在梯形图中只能使用输入继电器的触点，而不出现在它的线圈。而物理继电器触点的状态取决于其线圈中有无电流通过，在继电器控制电路中，若不接继电器线圈，只接其触点，则触点永远不会动作。

(2) 两种梯形图的区别

PLC 梯形图左右的两根线也叫母线，但与继电器控制电路的两根母线不同。继电器控制电路的母线与电源连接，其每一行(也称梯级)在满足一定条件时将通过两条母线形成电流通路，从而使电器动作；而 PLC 梯形图的母线并不接电源，它只表示每一个梯级的起始和终了，PLC 的每一个梯级中并没有实际的电流通过。通常说 PLC 的线圈接通了，这只不过是为了分析问题方便而假设的概念电流通路，而且概念电流只能从左向右流，这是 PLC 梯形图与继电器控制电路本质的区别。

(3) 实现控制功能手段的区别

继电器控制是靠改变电器间的硬接线来实现各种控制功能的，而 PLC 是通过程序来实现控制的。

图 1.7 是对应图 1.6(b) 的 PLC 外部接线。图中只画出了一部分输入和输出端子。0.00、0.01 等是输入端子，100.00、100.01 等是输出端子，输入和输出端子各有自己的公共端 COM。

当启动按钮 SB1 闭合时，0.00 输入端子对应的输入继电器线圈通电，它的触点相应动作；当停止按钮 SB2 闭合时，0.01 输入端子对应的输入继电器线圈通电，它的触点相应动作。当 100.00 输出端子对应的输出继电器线圈通电时，外部负载 KM 的线圈通电。根据上述关系，分析图 1.6(b) 启、停电动机的过程是：

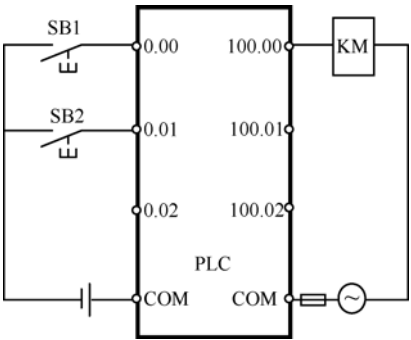


图 1.7 PLC 的外部接线

按下启动按钮 SB1，0.00 输入端子对应的输入继电器线圈通电，其常开触点 0.00 闭合。由于没有按动 SB2，所以常闭触点 0.01 处于闭合状态。因此输出继电器 100.00 线圈通电，使 KM 通电。KM 的主触点接在电动机的主电路中，于是电动机启动。释放启动按钮 SB1 后，由于 100.00 线圈通电，其常开触点 100.00 闭合起自锁作用。

在电动机运行过程中按下 SB2，0.01 输入端子对应的输入继电器线圈通电，其常闭触点 0.01 断开，输出继电器 100.00 线圈断电，使 KM 断电，于是电动机停转。

2. 语句表 IL 编程语言

语句表编程语言类似计算机的汇编语言，用助记符来表示各种指令的功能。对同样功能的指令，不同厂家的 PLC 使用的助记符一般不同。

对图 1.6(b) 的梯形图，其语句表为

LD	0.00	(常开触点 0.00 与左母线连接)
OR	100.00	(常开触点 100.00 与常开触点 0.00 相并联)
AND NOT	0.01	(串联一个常闭触点 0.01)
OUT	100.00	(输出到继电器 100.00)

指令语句是 PLC 用户程序的基础元素，多条语句的组合构成了语句表。一个复杂的控制功能是用较长的语句表来描述的。

语句表编程语言不如梯形图形象、直观，但是在使用简易编程器输入用户程序时，只能输入语句表程序。

3. 功能块 FB 编程语言

OMRON 推出的 CS1、CJ1 和 CP1 系列 PLC 和 CX-P(5.0 及其以上的版本)可直接支持用户定义的标准功能块编程。功能块是由完成特定功能的程序段组成的。功能块是一个包含标准处理功能的基本单元，该标准处理功能事先已定义好。用户可以将已定义好的功能块嵌入到梯形图程序中调用，同时设置执行功能块的输入/输出条件。功能块不包含实际地址，只有变量。用户可以在变量中设置地址或常数。

功能块编程的内容为：创建功能块和调用功能块。而创建一个功能块需要进行变量定义和算法设计。一旦在功能块中创建了标准编程并将其保存为文件，便可将功能块嵌入程序中并设置功能块 I/O 参数，进行反复使用，以达到简化程序的目的。

4. 结构文本 ST

随着 PLC 技术的发展，PLC 的高级功能越来越多，有时用梯形图来实现编程，会很不方便。为了方便用户使用 PLC 的数学运算、数据处理、图形显示、打印报表等功能，许多大中型或先进的 PLC 都配备了结构文本编程语言。

ST 语言是符合工业控制编程语言标准 IEC61131-3 的高级编程语言，用结构化的语句编写程序。在大、中型 PLC 控制系统中，常采用这种编程方式来表达控制系统中各变量的关系。

ST 语言类似于 Pascal 语言，但在表达式和语句的种类方面都做了简化。一个 ST 程序由多条语句组成，而一个语句是由变量和表达式组成。通过编写不同的语句建立不同的控制关系。表达式包含着运算符和操作数(变量或常数)。运算符包括算术运算、逻辑运算和比较运算等算法。语句可以分为赋值语句和控制语句。赋值语句将表达式计算的结果保存于变量中。控制语句包含选择语句、循环语句、条件语句等。

用户除了可以创建梯形图编程的 FB 之外，熟悉高级编程的技术人员还可以创建 ST 语言的 FB。CX-P 为熟悉不同编程语言的用户提供了不同的编程平台。懂得高级语言的用户编写好复杂的 ST 语言功能块后，其他用户甚至不熟悉 ST 语言的操作人员也可以使用该功能块，这就大大降低了对操作人员所使用编程语言的限制。另外，使用高级语言编程，可以轻松完成复杂的数学运算。若用梯形图语言编辑复杂的算术运算功能，需要编写很多条程序，占用更多的程序容量；而用 ST 语言来写，表达式就可以代替梯形图中的复杂程序，要方便得多。

5. 顺序功能图 SFC

顺序功能图是一种图形化的功能性说明语言，专用于描述工业顺序控制程序，使用它可以对具有并发、选择等复杂结构的系统进行编程。OMRON 的 CS1、CJ1 和 CP1(除 CP1E 外)等系列 PLC 支持 SFC 编程。

顺序功能图程序设计语言有如下特点：

① 以功能为主线，条理清楚，便于对程序操作的理解和沟通。

② 常用于系统的规模较大，程序关系较复杂的场合；对大型的程序，可分工设计，采用较为灵活的程序结构，可节省程序设计时间和调试时间。

③ 顺序功能图是由步构成的。在活动步的命令和操作被执行后，对活动步后的转换进行扫描，因此，整个程序的扫描时间较其他程序编制的程序扫描时间要大大缩短。

## 1.4 PLC 的工作方式和性能指标

### 1.4.1 PLC 的循环扫描工作方式

在继电器控制电路中，当某些梯级同时满足导通条件时，这些梯级中的继电器线圈会同时通电，也就是说，继电器控制电路是一种并行工作方式。PLC 是采用循环扫描的工作方式，在 PLC 执行用户程序时，CPU 对梯形图自上而下、自左向右地逐次进行扫描，程序的执行是按语句排列的先后顺序进行的。这样，PLC 梯形图中各线圈状态的变化在时间上是串行的，不会出现多个线圈同时改变状态的情况，这是 PLC 控制与继电器控制很重要的区别。

PLC 采用循环扫描的工作方式，它可以看做是一种由系统软件支持的扫描设备，不论用户程序运行与否，都周而复始地进行循环扫描，并执行系统程序规定的任务。每一个循环所经历的时间称为一个扫描周期。每个扫描周期又分为几个工作阶段，每个工作阶段完成不同的任务。

PLC 加电后首先进行初始化，然后进入循环扫描工作过程。完成一次扫描的任务及过程可归纳为 4 个工作阶段：公共处理→程序执行→I/O 刷新→外设端口服务，各阶段任务如下。

#### 1. 公共处理阶段

在每一次扫描开始之前，CPU 都要进行复位监视定时器、硬件检查，以及用户内存检查等操作。如果有异常情况，除了故障显示灯亮以外，还判断并显示故障性质。如果属于一般性故障，则只报警不停机，等待处理；如果属于严重故障，则停止 PLC 的运行。公共处理阶段所用的时间一般是固定的，不同机型的 PLC 有所差异。

#### 2. 程序执行阶段

在程序执行阶段，CPU 对用户程序按先上后下、先左后右的顺序逐条地进行扫描，即解释和执行。CPU 从输入映像寄存器(每个输入继电器对应一个输入映像寄存器)和元件映像寄存器(元件映像寄存器与各种内部继电器、输出继电器对应)中读取各继电器当前的状态，根据用户程序给出的逻辑关系进行逻辑运算，运算结果再写入元件映像寄存器中。

执行用户程序阶段的扫描时间不是固定的，其原因在于用户程序的指令条数多少不同，或指令种类不同。而且，在不同时段执行用户程序的扫描时间也不尽相同。

#### 3. I/O 刷新阶段

在 I/O 刷新阶段，CPU 要做两件事情。其一，从输入电路中读取各输入点的状态，并将此状态写入输入映像寄存器中，也就是刷新输入映像寄存器的内容。自此输入映像寄存器就与外界隔离，无论输入点的状态怎样变化，输入映像寄存器的内容都保持不变，一直到下一个扫描周期的 I/O 刷新阶段，才会写进新内容。这就是说，各输入映像寄存器的状态要保持一个扫描周期不变。其二，将所有输出继电器的元件映像寄存器的状态传送到相应的输出锁存电路中，再经输出电路的隔离和功率放大部分传送到 PLC 的输出端，驱动外部执行元件动作。

I/O 刷新阶段的时间长短取决于 I/O 点数的多少。

4. 外设端口服务阶段

在外设端口服务阶段中，CPU 完成与外设端口连接的外围设备的通信处理。

完成上述各阶段的处理后，又返回公共处理阶段，周而复始地进行扫描。

PLC 的循环扫描工作方式也为 PLC 提供了一条死循环自诊断功能。在 PLC 内部设置了一个监视定时器 WDT(看门狗)，WDT 定时时间可以使用默认值，也可由用户在 PLC 的系统设定区设定，其定时时间可设置为大于用户程序的扫描周期。PLC 在每个扫描周期的公共处理阶段将监视定时器复位。正常情况下，监视定时器不会动作。如果由于 CPU 内部故障使程序执行进入死循环，那么扫描周期将超过监视定时器的定时时间。这时监视定时器 WDT 动作使 PLC 运行停止，以提示用户排查故障。

图1.8描述了信号从输入端子到输出端子的传递过程。在 I/O 刷新阶段，CPU 从输入电路的输出端读出各输入点的状态，并将其写入输入映像寄存器中。在紧接着的下一个扫描周期用户程序执行阶段，CPU 从输入映像寄存器和元件映像寄存器中读出各继电器的状态，并根据此状态执行用户程序，执行结果再写入元件映像寄存器中。在 I/O 刷新阶段，将元件映像寄存器的状态写入输出锁存电路，再经输出电路传递到输出端子。

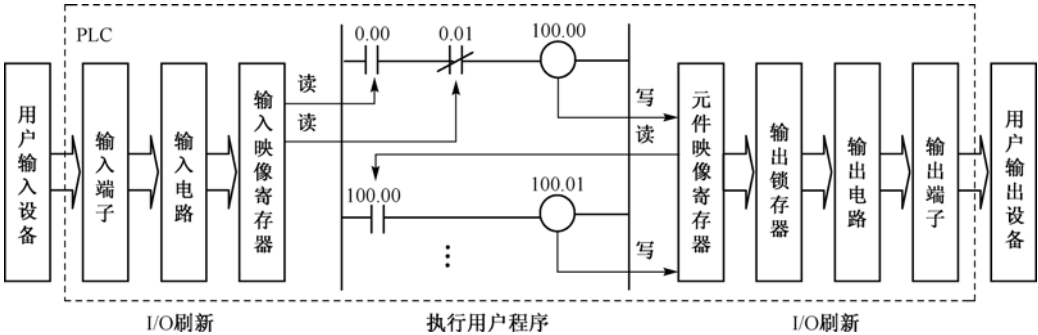


图 1.8 PLC 信号的传递过程

如图 1.8 所示，在某一个扫描周期里执行用户程序的具体过程：执行第一个梯级时，CPU 从输入映像寄存器中读出 0.00 号输入继电器的状态，设其为 1，再读出 0.01 号输入继电器的状态，设其为 0。由 0.00 和 0.01 的状态计算出 100.00 号继电器当前的状态是 1。若此前 100.00 的状态是 0，则 CPU 用当前的 1 去改写元件映像寄存器中 100.00 对应的位。下一步再执行第二个梯级，从元件映像寄存器中读出 100.00 号继电器的状态 1(即前一步存入的)，计算出 100.01 号继电器的状态是 1。若此前 100.01 的状态是 0，则 CPU 用当前的 1 去改写元件映像寄存器中 100.01 对应的位。本次扫描 I/O 刷新的结果是 100.00 为 1，100.01 为 1。

由上述分析可得出，执行用户程序的扫描阶段的特点如下：

- ① 在执行用户程序的过程中，输入映像寄存器的状态不变；
- ② 元件映像寄存器的内容随程序的执行在改变，前一步的计算结果随即作为下一步的计算条件，这一点与输入映像寄存器完全不同；
- ③ 程序的执行是由上而下进行，所以不同梯级中的继电器线圈及其触点的状态不可能同时改变，而继电器控制电路中，若几个梯级中的线圈都满足接通条件，那么这些线圈会同时通电。

PLC 的 I/O 滞后现象：

由于 PLC 采用循环扫描的工作方式，而且对输入和输出信号只在每个扫描周期的 I/O 刷新阶段集中输入、集中输出，所以必然会产生输出信号相对输入信号的滞后现象。扫描周期越长，滞后现象越严重。但是一般扫描周期只有十几毫秒，最多几十毫秒。在慢速控制系统中，可以认为输入信号一旦

变化就立即能进入输入映像寄存器中，其对应的输出信号也可以认为是及时的，而在要求快速响应的控制中就成了需要解决的问题。为了弥补此不足，PLC 设置了立即刷新指令，在程序中执行该指令，可随时对指定的 I/O 进行刷新，缩短了输出滞后输入的时间，提高了 I/O 响应速度。

对一般工业控制设备或输入信号变化较慢的系统来说，这种滞后现象是完全允许的。若需要输出对输入做出快速响应的场合，则可采用快速响应模块、高速计数模块，以及中断处理等措施来尽量减少滞后时间。

## 1.4.2 PLC 的主要性能指标

在描述 PLC 的性能时，经常用到以下术语：位(bit)、数字(digit)、字节(byte)及字(word)。位指二进制数的一位，仅有 1、0 两种取值。一个位对应 PLC 的一个继电器，某位的状态为 1 或 0，分别对应该继电器线圈得电(ON)或失电(OFF)。4 位二进制数构成一个数字，这个数字可以是 0000~1001(十进制数)，也可是 0000~1111(十六进制数)。2 个数字或 8 位二进制数构成 1 字节，2 字节构成一个字。在 PLC 术语中，字也称为通道。一个字含 16 位，或说一个通道含 16 个继电器。

PLC 的主要性能指标有以下几方面。

### 1. 存储容量

系统程序存放在系统程序存储器中。这里说的存储容量指的是用户程序存储器的容量。用户程序存储器容量决定了 PLC 可以容纳用户程序的长短，一般以字节为单位来计算。每 1024 字节为 1 KB(K 字)。中、小型 PLC 的存储容量一般在 20 KB 以下，大型 PLC 的存储容量可达到 256 KB 以上。PLC 程序存储容易常用步表示，某条指令占的步数是由 PLC 系统决定的，不可更改。一般基本指令占 1 步，应用指令占多步。CPIH PLC 的程序容量为 20 K 步。而 PLC 的数据存储容量常用 KB(K 字)表示。

### 2. 输入/输出点数

I/O 点数即 PLC 面板上的输入、输出端子的个数。I/O 点数越多，外部可接的输入器件和输出器件就越多，控制规模就越大。因此 I/O 点数是衡量 PLC 性能的重要指标之一。

### 3. 扫描速度

扫描速度是指 PLC 执行程序的速度，是衡量 PLC 性能的重要指标。一般以扫描 1 K 字所用的时间来衡量扫描速度。PLC 用户手册一般给出执行各条指令所用的时间，可以通过比较各种 PLC 执行相同的操作所用的时间，来衡量扫描速度的快慢。

### 4. 编程指令的种类和条数

编程指令的种类和条数是衡量 PLC 能力强弱的主要指标。编程指令种类及条数越多，其功能就越强，即处理能力、控制能力越强。

### 5. 存取器的种类和数量

内部器件包括各种继电器、计数器/定时器、数据存储器等。其种类越多，数量越大，存储各种信息的能力和控制功能越强。

### 6. 扩展能力

PLC 的扩展能力表现在以下几个方面。大部分 PLC 不仅可以连接基本 I/O 扩展单元进行 I/O 点数的扩展，还可以连接高功能 I/O 扩展单元和 CPU 扩展单元完成模拟量控制、位置和速度控制及通信联网等功能。扩展单元种类的多少和功能的强弱是衡量 PLC 产品水平高低的一个重要指标。

## 7. 通信能力

PLC 的通信能力表现在具有通信端口的个数、与各类外围设备实现连接和通信的能力,以及支持的通信协议有多少等。

## 1.5 PLC 产品及资料下载

从 20 世纪 70 年代初至今 40 年的时间里,PLC 生产已发展成为一个巨大的产业,主要厂商集中在欧美和日本。全世界有上百个 PLC 制造厂家,几家举足轻重的公司是日本的欧姆龙(OMRON)和三菱(Mitsubishi)公司,美国的罗克韦尔(Rockwell)自动化所属的 A-B(Allen & Bradley)公司,德国的西门子(Siemens)公司,法国的施耐德(Schneider)公司。欧、美的大型 PLC 的名气较大,日本则以高性价比的小型机著称。

### (1) 欧姆龙公司 PLC

日本的小型 PLC 很有特色,在小型机领域中颇具盛名。某些欧美中型或大型机才能实现的控制,用日本的小型机就可以解决。在开发较复杂的控制系统方面明显优于欧美的小型机,所以格外受用户欢迎。在中国 OMRON 的小型机产品销量居首位。

OMRON 公司的 PLC 产品,大、中、小型俱全。

小型机: CPM1A/CPM2A、CP1(CP1H/CP1L/CP1E)系列都属小型机。几年前,CPM1A/CPM2A 系列在我国的销量相当好,现在已被其升级产品、性价比更高的 CP1 系列所取代。

中型机: C200H、C200Hα、CJ1M 系列。其中 CJ1M 系列 PLC 是紧凑型的,具有更小的体积、更灵活的配置、更高的性能和更强的通信和网络功能,是近年畅销的高性能中型机。

大型机: C1000H/C2000H、CV/CVM1 系列、CS1 系列等。CS1 尽管是中型机 C200Hα 的后续机型,但就其功能而言,CS1 实际上是 OMRON 大、中型机的代表机型。CS1 推出后,原来需要大型机的场合 CS1 就胜任了,本文将其为大型机。

值得注意的是,小型机 CP1、中型机 CJ1 和大型机 CS1 的内部存储器设置趋向一致,存储器的种类结构几乎是一样的,但同一种内部存储器会稍有不同。例如,因为功能的不同,三种 PLC 的 CIO 区划分有差异;同一种内部存储器,在数量上,一般 CS1 要多于 CJ1 和 CP1。

目前欧姆龙主推的小、中和大机型为 CP1、CJ1 和 CS1 系列。它们具有相同的编程语言体系和几乎一样的系统内部存储器结构,其程序可以简单共享。

欧姆龙中国有限公司的工业自动化网站为<http://www.fa.omron.com.cn>,点击“软件及资料下载”进入下载中心,可以下载欧姆龙 PLC 的中英文资料和软件。另外,可以拨打欧姆龙中国有限公司的免费技术热线 400-820-4535(手机),询问技术问题。

### (2) 三菱公司 PLC

三菱公司的 PLC 也是较早进入中国市场的产品,其小型机有 FX<sub>0N</sub>、FX<sub>1S</sub>、FX<sub>2N</sub>、FX<sub>3U</sub>、FX<sub>3G</sub> 等。三菱公司的中、大型机为 Q 系列,有多种型号,如 Q<sub>02H</sub>、Q<sub>06H</sub>、Q<sub>12H</sub> 等。

在网站<http://www.mitsubishielectric-automation.cn/download/download.asp>上可以下载三菱 PLC 和其工控产品的资料。

### (3) 罗克韦尔自动化的 A-B(Allen & Bradley)公司 PLC

在进入我国的大型 PLC 中,美国 A-B 公司的产品几乎占到一半。罗克韦尔自动化公司的 PLC 产品主要有 PLC-5、MicroLogix、SLC500 可编程控制器和 ControlLogix 系列可编程控制器。

在网站<http://www.rockwellautomation.com.cn>上可以下载罗克韦尔 PLC 和其工控产品的中文资料。



#### (4) 德国西门子 PLC

德国西门子的电子产品以性能精良而久负盛名。在中、大型 PLC 产品领域与美国的 A-B 公司齐名。西门子目前销售机型为 S7 系列机,有 S7-200(小型)、S7-300(中型)及 S7-400(大型)。

在网站 <http://www.ad.siemens.com.cn> 上可以下载西门子 PLC 和其工控产品的资料。

#### (5) 施耐德公司 PLC

施耐德公司公司有三家 PLC 制造商: Modicon、Square D 与 Telemecanique(TE)公司。

在 [http://www.telemecanique.com/en/functions\\_discovery/function\\_5\\_11.htm](http://www.telemecanique.com/en/functions_discovery/function_5_11.htm) 上可以下载施耐德 PLC 的英文资料。在 <http://www.schneider-electric.com.cn/schneider/index.jsp> 的“下载中心”可以下载施耐德工控产品的中文资料。

网站 <http://www.plcs.net> 是一个 PLC 专业网站,在其“Learn PLCs”栏目中点击“Link”可以链接到全世界的 PLC 厂家网站和致力于 PLC 标准化的组织 The PLCopen 的网站。

在中国工控网的“PLC”专栏可以下载 PLC 的资料和软件,查询 PLC 的销售商和报价,在 PLC 论坛可以交流 PLC 应用中遇到的问题。

## 习 题

1. PLC 有哪些主要特点?
2. 比较 PLC 控制和继电器控制的优缺点。
3. 为什么称 PLC 的继电器是软继电器?和物理继电器相比,软继电器在使用上有何特点?
4. PLC 常采用哪几种编程语言?各有何特点?
5. 继电器控制和 PLC 控制的梯形图有何区别?
6. 整体式和组合式 PLC 主要由哪几个部分组成?简述各部分作用。
7. OMRON PLC 的编程工具通常有哪几种?各有何特点? CP1H 系列 PLC 的编程是怎样实现的?
8. CP1 系列 PLC 属于何种结构的 PLC?它与 CJ1/CS1 系列 PLC 相比又有哪些异同点?
9. 什么是 PLC 的扫描周期?其扫描过程分为哪几个阶段,各阶段完成什么任务?
10. PLC 控制与继电器控制的工作方式有何不同?执行用户程序的扫描阶段的特点是什么?
11. PLC 主要指标有哪些?各指标的意义是什么?

# 第 2 章 CP1 系列 PLC 的组成

OMRON 公司于 2005 年至 2009 年相继推出 CP1H、CP1L 和 CP1E 三款 CP1 系列小型整体 PLC。其中 2005 年推出的 CP1H PLC 是一款高性能的可编程控制器，引起了业界的广泛关注，CP1 系列 PLC 的结构是整体式的，CP1H 与 OMRON 公司上一代小型机 40 点的 CPM2A 尺寸相同，但其处理速度是 CPM2A 的 10 倍，它的内部存储器设置及指令系统与 CJ1/CS1 系列 PLC 几乎相同。CP1 系列 PLC 取消了手持编程器配置，没有通常的外设接口，取而代之的是 USB 端口，其中 CP1H 有串行通信功能端口两个，可自由选择 RS-232C 和 RS-485。

CP1 系列的三种 PLC CP1H/CP1L/CP1E 中，CP1H 是凝聚了多种功能的小型高功能 PLC，其 CPU 单元具有 20/40 点型；CP1L/CP1E 特点是丰富的 CPU 单元具有 10/14/20/30/40/60 点型；CP1E 具有经济、简单和高效特点。CP1H-XA40DR 型和 CP1E NA 型 PLC 有内置模拟量输入/输出功能，有些规格 PLC 可以通过扩展得到模拟量输入/输出功能。下面主要以 CP1H 为例叙述 CP1 系列 PLC 的主机组成规格和功能，其他 CP1 系列 PLC 在构成主要部分和编程方式与 CP1H PLC 类似，不同之处加以比较说明。初学者可以只看本章中 CP1H 主机规格和功能部分，CP1L 和 CP1E 规格部分在使用时作为选型参照。

## 2.1 CP1 系列 PLC 的基本构成

### 2.1.1 CP1 系列 PLC 的主机

#### 1. CP1 系列主机的规格

##### (1) CP1H 主机的规格

按使用电源的类型分，有 AC 型和 DC 型两种；按输出方式分，有继电器输出型和晶体管输出型两种。CP1H 的 CPU 单元有 3 种类型：X、XA 和 Y，其中 X 型为基本型，XA 型带内置模拟量输入/输出端子，Y 型带脉冲输入/输出专用端子，见表 2.1。

表 2.1 CP1H CPU 单元类型

类    型	型    号	输  出  形  式	电  源  电  压	I/O 点数	最大扩展 I/O 点数
X(基本型)	CP1H-X40DR-A	继电器	AC 100~240 V	40 点 24/16	连 7 台扩展单元，至 320 点
	CP1H-X40DT-D	晶体管(漏型)	DC 24 V		
	CP1H-X40DT1-D	晶体管(源型)	DC 24 V		
XA(带内置模拟量输入/出端子型)	CP1H-XA40DR-A	继电器	AC 100~240 V	40 点 24/16	
	CP1H-XA40DT-D	晶体管(漏型)	DC 24 V		
	CP1H-XA40DT1-D	晶体管(源型)	DC 24 V		
Y(带脉冲输入/出专用端子型)	CP1H-Y20DT-D	晶体管(漏型)	DC 24 V	2 点 12/8	连 7 台扩展单元，至 300 点

注：CP1 的基本 I/O 扩展单元可以连接 CP1W(或 CPM1A)系列扩展 I/O 单元。

##### (2) CP1L 主机的规格

CP1L 的 CPU 单元有两种类型：M 型和 L 型，见表 2.2。

表 2.2 CP1L CPU 单元类型

类 型	型 号	输 出 形 式	电 源 电 压	I/O 点数	最大扩展 I/O 点数
M 型	CP1L-M60DR-A (或-D)	继电器	末尾-A: AC 100~240 V; 末尾-D: DC 24 V	60 点 36/24	连 3 台扩展单元, 至 180 点
	CP1L-M60DT-A (或-D)	晶体管(漏型)			
	CP1L-M60DT1-D	晶体管(源型)		40 点 24/16	连 3 台扩展单元, 至 160 点
	CP1L-M40DR-A (或-D)	继电器			
	CP1L-M40DT-A (或-D)	晶体管(漏型)			
	CP1L-M40DT1-D	晶体管(源型)			
L 型	CP1L-M30DR-A (或-D)	继电器	末尾-A: AC 100~240 V; 末尾-D: DC 24 V	30 点 18/12	连 3 台扩展单元, 至 150 点
	CP1L-M30DT-A (或-D)	晶体管(漏型)			
	CP1L-M30DT1-D	晶体管(源型)		20 点 12/8	连 1 台扩展单元, 至 60 点
	CP1L-L14DR-A (或-D)	继电器			
	CP1L-L14DT-A (或-D)	晶体管(漏型)			
	CP1L-L14DT1-D	晶体管(源型)			
	CP1L-L10DR-A (或-D)	继电器		14 点 8/6	连 1 台扩展单元, 至 54 点
	CP1L-L10DT-A (或-D)	晶体管(漏型)			
	CP1L-L10DT1-D	晶体管(源型)			
	CP1L-L10DR-A (或-D)	继电器	末尾-A: AC 100~240 V; 末尾-D: DC 24 V	10 点 6/4	不能连接扩展单元
	CP1L-L10DT-A (或-D)	晶体管(漏型)			
	CP1L-L10DT1-D	晶体管(源型)			

注: CP1 的基本 I/O 扩展单元可以连接 CP1W (或 CPM1A) 系列扩展 I/O 单元。

(3) CP1E 主机的规格

CP1E 的 CPU 单元有 3 种类型: E 型、N 型和 NA 型, 见表 2.3。

表 2.3 CP1E CPU 单元类型

类 型	型 号	输 出 形 式	电 源 电 压	I/O 点数	最大扩展 I/O 点数
E 型 (基本型)	CP1E-E40DR-A	继电器	末尾-A: AC 100~240 V ; 末尾-D: DC 24 V	40 点 24/16	连 3 台扩展单元, 至 160 点
	CP1E-E30DR-A	继电器		30 点 18/12	连 3 台扩展单元, 至 150 点
	CP1E-E20DR-A	继电器		20 点 12/8	不能连接扩展单元
	CP1E-E14DR-A	继电器		14 点 8/6	
	CP1E-E10DR-A (或-D)	继电器		10 点 6/4	
	CP1E-E10DT-A (或-D)	晶体管(漏型)			
	CP1E-E10DT1-A (或-D)	晶体管(源型)			
N 型 (应用型)	CP1E-N60DR-A (或-D)	继电器	末尾-A: AC 100~240 V ; 末尾-D: DC 24 V	60 点 36/24	连 3 台扩展单元, 至 180 点
	CP1E-N60DT-A (或-D)	晶体管(漏型)		40 点 24/16	连 3 台扩展单元, 至 160 点
	CP1E-N60DT1-A (或-D)	晶体管(源型)			
	CP1E-N40DR-A (或-D)	继电器			
	CP1E-N40DT-A (或-D)	晶体管(漏型)			
	CP1E-N40DT1-A (或-D)	晶体管(源型)			
	CP1E-N30DR-A (或-D)	继电器		20 点 12/8	不能连接扩展单元
	CP1E-N30DT-A (或-D)	晶体管(漏型)			
	CP1E-N30DT1-A (或-D)	晶体管(源型)			
	CP1E-N20DR-A (或-D)	继电器		14 点 8/6	
CP1E-N20DT-A (或-D)	晶体管(漏型)				
CP1E-N20DT1-A (或-D)	晶体管(源型)				
NA 型 (增强型)	CP1E-N14DR-A (或-D)	继电器	-A: AC 100~240 V ; -D: DC 24 V	20 点 12/8	连 3 台扩展单元, 至 140 点
	CP1E-N14DT-A (或-D)	晶体管(漏型)			
	CP1E-N14DT1-A (或-D)	晶体管(源型)			

注: CP1 的基本 I/O 扩展单元可以连接 CP1W (或 CPM1A) 系列扩展 I/O 单元。

2. CP1H 主机的面板结构和基本功能

CP1 系列 PLC 的结构是类似的，仅以 CP1H-XA40DR-A 型 PLC 为例介绍主机面板结构和基本功能。其他 CP1 系列 PLC 的结构可以查找 OMRON 公司的相关手册。

CP1H 属整体式结构，除中央处理单元(CPU)、存储器、输入单元、输出单元、电源外，还有 USB 端口、通信端口，还可以加装扩展单元等。下面介绍日本欧姆龙公司整体式的 CP1H-XA40DR-A 型 PLC 主机面板的布置及各个接线端子和接口的作用，面板结构如图 2.1 所示。

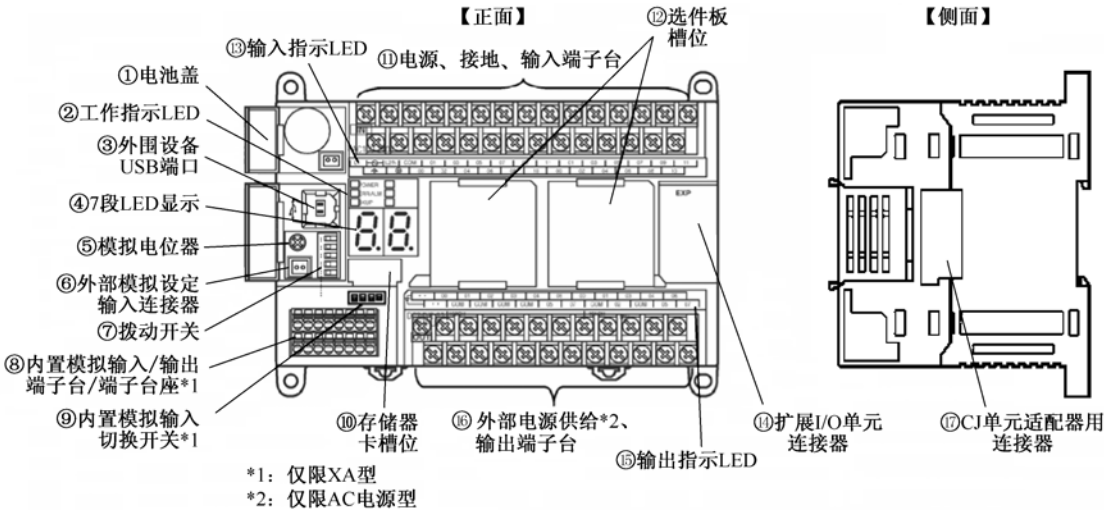


图 2.1 CP1H-XA40DR-A 型 PLC 主机面板

现将 CP1H 各组成部分功能逐一简要说明如下：

- ① 电池盖。打开盖可将电池放入，用做 RAM 的后备电源。
- ② 工作指示 LED。指示 CP1H 的工作状态。主机面板的中部有 6 个工作状态显示 LED，如图 2.2 所示，其作用见表 2.4。

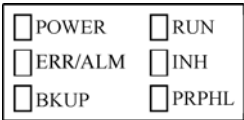


图 2.2 工作指示 LED

表 2.4 工作指示 LED 的含义

名 称	状 态	含 义
POWER(绿) 电源接通或断开指示	灯亮	通电
	灯灭	未通电
RUN(绿) PLC 工作状态指示	灯亮	CP1H 正在运行或监视模式下执行程序
	灯灭	CP1H 处在编程状态或运行异常时灭
ERR/ALM(红) 错误指示	灯亮	严重错误指示。发生运行停止异常(包含 FAL 指令执行)，或发生硬件异常(WDT 异常)时，CP1H 停止运行，所有的输出都切断
	闪烁	警告性错误指示。发生异常 CP1H 继续运行(包含 FAL 指令执行)
	灯灭	正常
INH(黄) 输出禁止指示	灯亮	输出禁止特殊辅助继电器(A500.15)为 ON 时灯亮，所有输出都切断
	灯灭	正常
BKUP(黄) 内置闪存访问指示	灯亮	正在向内置闪存(备份存储器)写入用户程序、参数、数据或访问中。此外 PLC 的电源变 ON 时，用户程序、参数、数据复位过程中灯也亮
	灯灭	上述情况以外
PRPHL(黄) USB 端口通信指示	闪烁	外围设备 USB 端口处于通信中时
	灯灭	不通信时

- ③ 外围设备 USB 端口。与计算机连接，由安装在上位机的软件 CX-P 对 PLC 进行编程及监视。
- ④ 7 段 LED 显示。在 2 位的 7 段 LED 上显示 CP1H CPU 单元的异常信息及模拟电位器操作时的当前值等 CPU 单元的状态。CP1L/CP1E 没有此功能。
- ⑤ 模拟电位器。通过旋转电位器，可使 A642 CH 的值在 0~255 范围内任意变更。
- ⑥ 外部模拟设定输入连接器。通过外部施加 0~10 V 电压，可将 A643 CH 的值在 0~255 范围内任意变更。
- ⑦ 拨动开关。6 个拨动开关的作用见表 2.5。

表 2.5 拨动开关的作用

No.	设 定	设 定 内 容	用 途	初 始 值
SW1	ON	不可写入用户存储器	在需要防止由外围工具(CX-Programmer)导致的不慎改写程序的情况下使用	OFF
	OFF	可写入用户存储器		
SW2	ON	电源为 ON 时,将存于存储盒的内容自动传送到 CPU	在电源为 ON 时,可将保存在存储盒内的程序、数据内存(存储)、参数自动传送到 CPU 单元	OFF
	OFF	不执行		
SW3	—	未使用	—	OFF
SW4	ON	在用工具总线情况下使用	需要通过工具总线来使用选件板槽位 1 上安装的串行通信端口时置于 ON	OFF
	OFF	根据 PLC 系统设定		
SW5	ON	在用工具总线情况下使用	需要通过工具总线来使用选件板槽位 2 上安装的串行通信端口时置于 ON	OFF
	OFF	根据 PLC 系统设定		
SW6	ON	A395.12 为 ON	通过 SW6 将继电器 A395.12 置于 ON 或 OFF	OFF
	OFF	A395.12 为 OFF		

⑧ 内置模拟输入/输出端子台(仅限 XA 型)。模拟输入 4 点、模拟输出 2 点。模拟量输入/输出端子台排列及引脚功能如图 2.3 所示。详细使用将在后面第 7 章介绍。

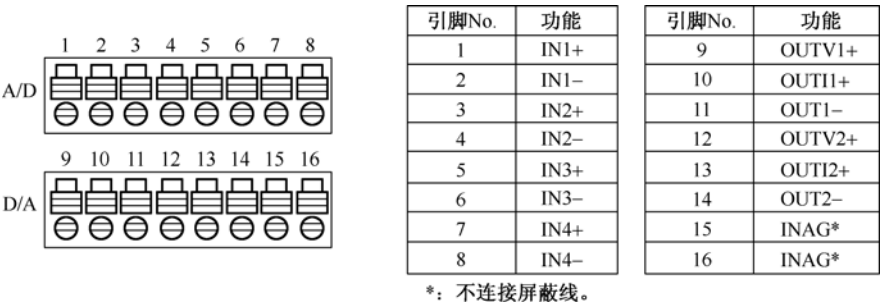
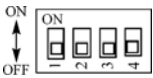


图 2.3 模拟量输入/输出端子台排列及引脚功能

⑨ 内置模拟输入切换开关(仅限 XA 型)。将各模拟输入在电压输入下使用还是电流输入下使用间切换。切换开关的 1~4 分别用来设定模拟输入 1~模拟输入 4 的电流或电压输入。某一切换开关为 ON，其相应模拟输入为电流输入；OFF 时为电压输入。出厂设定为电压输入。如图 2.4 所示为内置模拟输入切换开关。



- ⑩ 存储器卡槽位。安装 CP1W-ME05M(512 KB)存储器卡。可将 CP1H CPU 单元的梯形图程序、参数、数据内存(DM)等传送并保存到存储器卡。
- ⑪ 电源、接地、输入端子台。电源、接地、输入端子台作用见表 2.6。

表 2.6 电源、接地、输入端子台作用

名 称	作 用
电源端子	供给电源(AC 100~240 V 或 DC 24 V)
接地端子	功能接地: 为了强化抗干扰性、防止电击, 必须接地 保护接地: 为了防止触电, 必须进行 D 种接地(第 3 种接地)
输入端子	连接输入设备。内置 24 点输入端子: 00.00~00.11, 01.00~01.11

⑫ 选件板槽位。槽位 1 和槽位 2 上可分别选装 RS-232C 口 CP1W-CIF01、RS-422A/485 口 CP1W-CIF11 或 LCD 显示操作面板 CP1W-DAM01。

⑬ 输入指示 LED。内置 24 点输入端子 00.00~00.11, 01.00~01.11 对应的指示灯。输入端子的接点为 ON 时, 指示灯亮。

⑭ 扩展 I/O 单元连接器。可连接 CP1W 或 CPM1A 系列的扩展 I/O 单元(40 点、输出型 32 点、20 点、输出型 16 点、输入型 8 点和输出型 8 点)及扩展单元(模拟输入/输出单元、温度传感器单元、CompoBus/S I/O 连接单元、DeviceNet I/O 连接单元), 最大为 7 台。

⑮ 输出指示 LED。内置 16 点输出端子 100.00~100.07 和 101.00~101.07 对应的指示灯。输出端子的接点为 ON 时, 指示灯亮。

⑯ 外部电源供给、输出端子台。对外部提供 DC 24 V, 最大 300 mA 的电源。可作为输入设备或现场传感器的服务电源。

⑰ CJ 单元适配器用连接器。位于 CP1H CPU 单元的侧面, 要通过 CJ 单元适配器 CP1W-EXT01, 连接 CJ 系列特殊 I/O 单元或 CPU 总线单元, 最多两个单元。但是注意不可以连接 CJ 系列的基本 I/O 单元。CP1L/CP1E 没有此功能。

3. CP1H、CP1L 和 CP1E 功能比较

CP1 系列 PLC 的结构是类似的, 是整体式小型 PLC, 在主要构成、编程方式等方面也类似。但不型号和规格的 PLC 的串行通信、脉冲输出和计数器等功能有差别, 其功能比较见表 2.7。

表 2.7 CP1 系列 PLC 功能比较

名 称	CP1H			CP1L		CP1E	
型号	XA 型 40 点	X 型 40 点	Y 型 20 点	M 型 60/40 /30 点	L 型 20/14/ 10 点	NA 型 20 点、N 型 60/40/30/20/14 点	E 型 40/30/20/ 14/10 点
脉冲输出 功能	100 kHz×4 轴		1 MHz×2 轴、 100 kHz×2 轴	100 kHz×2 轴		100 kHz×2 轴	—
计数器功能	增量输入 100 kHz×4 轴 加/减法输入 100 kHz×4 轴 脉冲+方向输入 100 kHz×4 轴 相位差输入 50 kHz×4 轴		增量输入 1 MHz×4 轴 加/减输入 1 MHz×4 轴 脉冲+方向输入 1 MHz×4 轴 相位差输入 50 kHz×4 轴	增量输入 100 kHz × 4 轴 加/减输入 100 kHz × 2 轴 脉冲+方向输入 100 kHz × 2 轴 相位差输入 50 kHz×2 轴		增量输入 100 kHz×2 轴、10 kHz×4 轴 加/减输入 100 kHz×1 轴、10 kHz×1 轴 脉冲+方向输入 100 kHz×2 轴 相位差输入 50 kHz×1 轴、5 kHz×1 轴	增量输入 10 kHz×6 轴(10 点×5 轴) 加/减输入 10 kHz×2 轴 脉冲+方向输入 10 kHz×2 轴 相位差输入 5 kHz×2 轴
串行通 信功能	可选装 2 个串行端口 RS232C 或 RS422/485			选装 2 个 RS 232C 或 RS422 /485	选装 1 个 RS 232C 或 RS422 /485(10 点无)	N 型 20/14 点: 1 个 RS232C 60/40/30 点和 NA 型 20 点: 1 个 RS232+选装 1 个 RS232C 或 RS422/485	—
以太网 通信	装 1 个以太网端口			装 1 个以太网 端口	装 1 个以太网 端口(10 点无)	60/40/30 点 NA 型 20 点: 装 1 个以太网端口	—
USB 端口	有	有	有	有	有	有	有

(续表)

名 称	CP1H			CP1L		CP1E	
内置模拟量 I/O	输入 4/ 输出 2 通道	—	—	—	—	仅限 NA 型 输入 2/输出 1 通道	—
内存盒	有	有	有	有	有	有	有
7 段 LED 显示	有	有	有	—	—	—	—
LCD 显示板	可选装	可	可选装	可选装	可选装 (10 点无)	—	—
功能块	有	有	有	有	有	有	有
变频器定位	—	—	—	有	有	有	有
7 段显示	有	有	有	—	—	—	—
程序容量	20 K 步			10 K 步	5 K 步	8 K 步	2 K 步
数据存储容量	32 K 字			32 K 字	10 K 字	8 K 字	2 K 字
扫描速度	0.10 μs LD 指令, 0.3 μs MOV 指令			0.55 μs LD 指令, 4.1 μs MOV 指令		1.19 μs LD 指令, 7.9 μs MOV 指令	

4. CP1 系列 PLC 外部设备

PLC 的外部设备通常包括个人计算机(含专用适配卡和组态软件)、编程设备、显示终端,以及可编程终端、打印机、数据访问设备等。

(1) 编程设备

常见的 PLC 编程设备包括手持编程器(也称简易编程器)和计算机辅助编程。CP1 系列 PLC 只能使用计算机辅助编程。在 PC 上装上 CX-P 编程软件,用专用电缆将 PC 的 USB 端口与 PLC 的 USB 端口相连接,或用通信电缆将 PC 的 RS-232C 端口与 CP1 的 RS-232C 端口相连接。用户通过 CX-Programmer(简称 CX-P)编程软件在计算机上编程,然后下载到 CP1 系列 PLC 中。有关编程软件 CX-P 的使用方法详见第 4 章。

(2) 可编程终端

可编程终端(Programmable Terminals, PT)又称工业级人机界面(Human Machine Interface, HMI), PT 与 PLC 用通信线缆连接后,通过薄膜键盘或屏幕上的触摸键,可向 PLC 输入数据。在普通个人计算机(Personal Computer, PC)上使用厂家提供的支持软件,通过调用各种控件,如按钮、数值输入、指示灯、数值显示、文字显示、信息显示、条状图、曲线图、XY 图、仪表、动态图、图形元素、静态文字、静态图形及通用元件,可以制作数百种生动、丰富的画面。完成编译后,通过连接 PC 的 RS-232C 端口和 PT 的 RS-232C 端口的通信线缆,将控制画面由 PC 下载到 PT,此后,PT 就可以脱离 PC 而独立运行。使用 PT,可取消传统的控制面板,从而简化了硬件,减少了接线,也就节省了 PLC 的 I/O 单元。

PT 的第 2 个通信端口(RS-232C)与 PLC 的通信端口(RS-232C)通过通信线缆相连接,此后,可使用 PT 对 PLC 控制系统进行操作并显示各种信息。

2.1.2 CP1 系列 PLC 的其他功能

1. 内置模拟量输入/输出功能

CP1H-XA40DR-A 型 CPU 单元最显著的特点是它的内置模拟量输入/输出功能(XA 型的 CP1H CPU 单元中内置模拟输入 4 点/输出 2 点),内置模拟量输入/输出功能见表 2.8。另外 CP1E NA 型 20 点 CPU 单元中有内置模拟量输入 2 点/输出 1 点。

表 2.8 CP1H 内置量模拟输入/输出功能

项 目		电压输入/输出 <sup>①</sup>	电流输入/输出 <sup>①</sup>
模拟输入	模拟输入点数	4 点(占用 200~203CH, 共 4CH)	
	输入信号量程	0~5 V、1~5 V、0~10 V、-10~10 V	0~20 mA、4~20 mA
	最大额定输入	± 15 V	± 30 mA
	外部输入阻抗	1 MΩ以上	约 250 Ω
	分辨率	1/6000 或 1/12 000 (FS: 满量程) <sup>②</sup>	
	综合精度	25℃ ± 0.3%FS/0~55℃ ± 0.6%FS	25℃ ± 0.4%FS/ 0~55℃ ± 0.8%FS
	A/D 转换数据	-10~10 V 时: 满量程值 F448 (E890)~0BB8 (1770) Hex 上述以外: 满量程值 0000~1770 (2EE0) Hex	
	平均化处理	有(通过 PLC 系统设定来设定各输入)	
	断线检测功能	有(断线时的值 8000 Hex)	
模拟输出	模拟输出点数	2 点(占用 210、211CH, 共 2CH)	
	输出信号量程	0~5 V、1~5 V、0~10 V、-10~10 V	0~20 mA、4~20 mA
	外部输出允许负载电阻	1 kΩ以上	600 Ω以下
	外部输出阻抗	0.5 Ω以下	—
	分辨率	1/6000 或 1/12000 (FS: 满量程) <sup>②</sup>	
	综合精度	25℃ ± 0.4%FS / 0~55℃ ± 0.8%FS	
	D/A 转换数据	-10~10 V 时: 满量程值 F448 (E890)~0BB8 (1770) Hex 上述以外: 满量程值 0000~1770 (2EE0) Hex	
转换时间		1 ms/点 <sup>③</sup>	
隔离方式		模拟输入/输出与内部电路间为光电耦合器隔离(但模拟输入/输出间为不隔离)	

注: ① 电压输入/电流输入的切换由内置模拟输入切换开关来完成(出厂时设定为电压输入);  
② 分辨率 1/6000、1/12000 的切换由 PLC 系统设定来进行, 所有的输入/输出通道只能用同一个分辨率设定;  
③ 合计转换时间为所使用的点数的转换时间的合计, 使用模拟输入 4 点和模拟输出 2 点时为 6 ms。

2. 中断功能和快速响应功能

CP1H 的 CPU 单元, 通常周期性重复“公共处理→程序执行→I/O 刷新→外设端口服务”4 个工作阶段, 执行周期性任务。与此不同, 根据特定要求的发生, 可以在该周期执行任务的中途中断, 使其执行特定的程序, 这称为中断功能。

PLC 的输出对输入的响应速度受扫描周期的影响, 在某些特殊情况下可能使一些瞬间的输入信号被遗漏。为了防止发生这种情况, CP1H 系列 PLC 中设计了快速响应输入功能。有了这个功能, PLC 可以不受扫描周期的影响随时接收最小脉冲信号宽度为 30 μs 的瞬间脉冲。X/XA 型的情况下最大可使用 8 点, Y 型的情况下最大可使用 6 点。其功能规格见表 2.9。

表 2.9 输入中断和快速响应输入(X/XA 型)

项 目		规 格
中断输入和快速响应输入点数		共用内置输入端子, 共 8 点
输入中断	输入中断直接模式	在输入信号的上升沿或下降沿, 中断 CPU 单元的循环程序, 并且执行相应 I/O 中断任务
	输入中断计数器模式	输入信号的上升(沿)或下降(沿)的次数被增量或减量计数, 当计数值达到时, 相应的中断任务开始执行, 输入响应频率为 5 kHz 以下
快速响应输入		小于循环时间(最小为 30 μs)的信号可作为 ON 信号的一个周期处理



3. 高速计数器功能

CP1H 系列 PLC 有 4 个高速计数器，CP1H 高速计数器的输入模式有 4 种：递增模式、相位差输入模式、增/减(加/减)模式和脉冲+方向模式。高速计数器计数模式有两种：线性模式和环形模式。

使用高速计数器时，部分内容要预先在 CX-P 编程软件上进行设置，否则高速计数器是不工作的。高速计数器的规格见表 2.10。

表 2.10 高速计数器的规格

项 目		规 格			
高速计数器点数		4 点(高速计数器 0~3)			
输入模式		递增模式	相位差输入模式	增/减(加/减)模式	脉冲+方向模式
响应频率	24VDC 输入	100 kHz	50 kHz	100 kHz	100 kHz
计数模式		线形模式或循环模式			
计数范围		线形模式：80000000~7FFFFFFF Hex 循环模式：00000000~环形计数器设定值			
高速计数器当前值 存储通道		高速计数器 0：A271(高 4 位)/A270(低 4 位) 高速计数器 1：A273(高 4 位)/A272(低 4 位) 高速计数器 2：A317(高 4 位)/A316(低 4 位) 高速计数器 3：A319(高 4 位)/A318(低 4 位) 这些值用于与目标值比较表或区域比较表中的值进行比较			
控制方式	目标值比较	最多可登录 48 个目标和中断任务编号			
	区域比较	最多可登录 8 个高限、低限和中断任务编号			
计数器复位方式		① Z 相信号+软件复位：当复位位为 ON 及 Z 相输入转为 ON 时，计数器复位 ② 软件复位：当复位位为 ON 时，计数器复位 ③ Z 相信号+软件复位(继续) ④ 软件复位(继续) 复位位：A531.00(高速计数器 0)、A531.01(高速计数器 1)、A531.02(高速计数器 2)、A531.03(高速计数器 3)			

4. 脉冲输出功能

可从 CPU 单元内置输出中发出固定占空比脉冲输出信号,并通过脉冲输入的伺服电动机驱动器进行定位/速度控制。脉冲输出功能见表 2.11 和表 2.12。

表 2.11 脉冲输出功能

项 目	规 格
输出模式	连续模式(速度控制用)或单独模式(位置控制用)
输出频率	X/XA 型：1 Hz~100 kHz 4 点(脉冲输出 0、1、2、3) Y 型：1 Hz~1 MHz 2 点(脉冲输出 0、1)，1 Hz~100 kHz 2 点(脉冲输出 2、3)
频率加速/减速	X/XA/Y 型：1~65 535Hz(每 4 ms)，以 1 Hz 为单位设定每 4 ms 加速或减速速率
指令执行中改变设定值	可以改变目标频率、加速/减速速率及目标位置的变更
脉冲输出方式	CW/CCW(顺时针/逆时针)或脉冲+方向，固定占空比 50%
输出脉冲数	相对坐标范围：00000000~7FFFFFFF 十六进制(2 147 483 647) 绝对坐标范围：80000000~7FFFFFFF 十六进制(-2 147 483 648~2 147 483 647)
原点搜索/复位	ORG (ORIGIN SEARCH)：用于执行原点搜索或按设定值复位
定位及速度控制指令	PLS2(pulse output)：用于分别设定加速和减速速率进行梯形定位控制的输出脉冲 PULS(set pulses)：用于设定输出脉冲数 SPED(speed output)：用于无加速或减速作用的输出脉冲 ACC(acceleration control)：用于控制加速/减速速率 INI(mode control)：用于停止脉冲输出
脉冲输出当前值存储区	脉冲输出 0：A277(高 4 位数)/A276(低 4 位数) 脉冲输出 1：A279(高 4 位数)/A278(低 4 位数) 脉冲输出 2：A323(高 4 位数)/A322(低 4 位数) 脉冲输出 3：A325(高 4 位数)/A324(低 4 位数) I/O 刷新时被更新

表 2.12 占空比可变的脉冲(PWM)输出功能

项 目	规 格
占空比	0.0~100.0%，设定单位 0.1%
频率	0.1~6553.5 Hz，设定单位 0.1 Hz
PWM 用指令	PWM(可变占空比的脉冲)：用于输出指定占空比的脉冲
输出点数	2 点。PWM 输出 0，位地址为 101.00；PWM 输出 1，位地址为 101.01

2.2 CP1 系列 PLC 的扩展单元

CP1H CPU 单元扩展时, 可以通过 I/O 单元连接器连接最多 7 台 CP1W(或 CPM1A) 系列的各种扩展单元, 还可以通过 CJ 单元适配器 CP1W-EXT01 连接两台 CJ 系列的高功能单元(特殊 I/O 单元、CPU 总线单元), 但不可以连接 CJ 的基本 I/O 单元。CP1L 和 CP1E CPU 单元只可以通过 I/O 单元连接器连接最多 3 台 CP1W(或 CPM1A) 系列的各种扩展单元, 不能连接 CJ 系列的高功能单元。

2.2.1 连接 CP1W 和 CPM1A 系列扩展单元

CPW 扩展单元是欧姆龙 CP1 系列的扩展单元, CPM1A 扩展单元是 CPM 系列 PLC 的扩展单元。

欧姆龙 CP1 系列 PLC 为整体式结构, 内置开关量 I/O 点数为 10~60 点。当 I/O 点数不够用或需要模拟量 I/O 时, 可连接扩展 I/O 单元, CP1H 最多可连接 7 台 CP1W(或 CPM1A) 系列扩展 I/O 单元, 最大输入/输出点数可达 320 点。CP1 系列 PLC 的基本 I/O 扩展单元见表 2.13、特殊 I/O 扩展单元见表 2.14。开关量扩展 I/O 单元有 40 点型、32 点输出型、20 点型、16 点输出型、8 点输入型和 8 点输出型。图2.5 为 CP1 系列 PLC 连接 CP1W(或 CPM1A) 系列扩展单元的配置图, 其中, 图2.5 (a) 为常用的简单配置, 图2.5 (b) 为使用 I/O 连接电缆 CP1W-CN811 的配置。电缆可延长 0.8 m, 并可两段并行。

表 2.13 CP1W 和 CPM1A 基本 I/O 扩展单元

型 号	类型及占用通道数(CH)	输 出 形 式	占用通道数(输入/输出)
CP1W-40EDR CPM1A-40EDR	40 点 输入：24 点(2CH) 输出：16 点(2CH)	继电器	2/2
CP1W-40EDT CPM1A-40EDT		晶体管(漏型)	
CP1W-40EDT1 CPM1A-40EDT1		晶体管(源型)	
CP1W-32ER CP1W-32ET CP1W-32ET1	32 点 输出：32 点(4CH)	继电器 晶体管(漏型) 晶体管(源型)	无/4
CP1W-20EDR1 CPM1A-20EDR1	20 点 输入：12 点(1CH) 输出：8 点(1CH)	继电器	1/1
CP1W-20EDT CPM1A-20EDT		晶体管(漏型)	
CP1W-20EDT1 CPM1A-20EDT1		晶体管(源型)	
CP1W-16ER CPM1A-16ER CP1W-16ET CP1W-16ET1	16 点 输出：16 点(2CH)	继电器 继电器 晶体管(漏型) 晶体管(源型)	无/2
CP1W-8ED CPM1A-8ED	8 点 输入：8 点(1CH)	无	1/无
CP1W-8ER CPM1A-8ER	8 点 输出：8 点(1CH)	继电器	无/1
CP1W-8ET CPM1A-8ET		晶体管(漏型)	
CP1W-8ET1 CPM1A-8ET1		晶体管(源型)	

表 2.14 CP1W 和 CPM1A 特殊 I/O 扩展单元

名 称	型 号	规 格			占用通道数 (输入/输出)
模拟量输入单元	CP1W-AD041 CPM1A-AD041	模拟输入：4 点	电压：0～5 V/1～5 V/0～10 V/－10～+10 V 电流：0～20 mA/4～20 mA	分辨率 6000	4/无
模拟量输出单元	CP1W-DA041 CPM1A-DA041	模拟输出：4 点	电压：1～5 V/0～10 V/－10～+10 V 电流：0～20 mA/4～20 mA	分辨率 6000	无/ 4
模拟量 I/O 单元	CPM1A-MAD01	模拟输入：2 点	电压：0～10 V/1～5 V 电流：4～20 mA	分辨率 256	2/1
		模拟输出：1 点	电压：0～10 V/－10～+10 V 电流：4～20 mA		
	CP1W-MAD11 CPM1A-MAD11	模拟输入：2 点	电压：0～5 V/1～5 V/ 0～10 V/－10～+10 V 电流：0～20 mA/4～20 mA	分辨率 6000	
		模拟输出：1 点	电压：1～5 V/0～10 V/－10～+10 V 电流：0～20 mA/4～20 mA		
温度传感器单元	CP1W-TS001 CPM1A-TS001	输入：2 点	热电偶输入 K, J 之间选一		2/无
	CP1W-TS002 CPM1A-TS002	输入：4 点			4/无
	CP1W-TS101 CPM1A-TS101	输入：2 点	铂热电阻输入		2/无
	CP1W-TS102 CPM1A-TS102	输入：4 点	Pt100, JPt100 之间选一		4/无
DeviceNet I/O 链接单元	CPM1A-DRT21	DeviceNet 从站 I/O 点数：输入 32 点, 输出 32 点			2/2
CompoBus/S I/O 链接单元	CP1W-SRT21 CPM1A-SRT21	CompoBus/S 从站 I/O 点数：输入 8 点, 输出 8 点			1/1

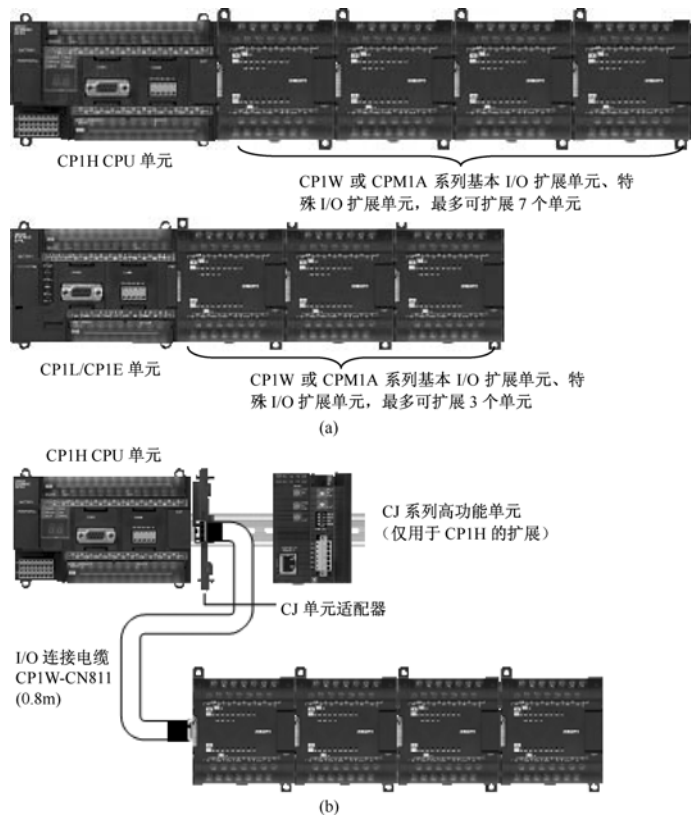


图 2.5 CP1H 的扩展配置

图2.6为 CP1H 连接 CPM1A 系列扩展单元时的 I/O 通道分配。

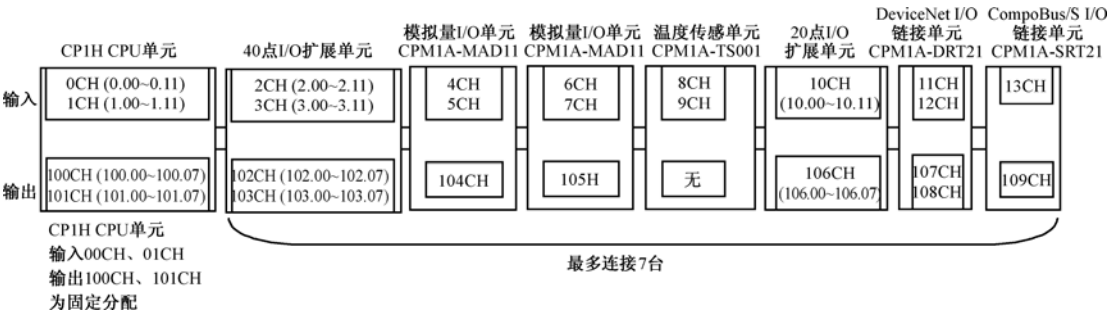


图 2.6 CP1H 连接 CPM1A 系列扩展单元的 I/O 通道分配

2.2.2 连接 CJ1 系列高性能单元

图2.7为 CP1H CPU 单元连接 CJ1 系列高性能单元的配置图，最多可连接两台。为了连接，CJ1 单元适配器 CP1W-EXT01 和端板 CJ1W-TER01 是必需的。这种配置下，CP1H 可扩展网络通信和协议宏等串行通信功能。CP1L 和 CP1E 不能连接 CJ 系列的高性能扩展单元。

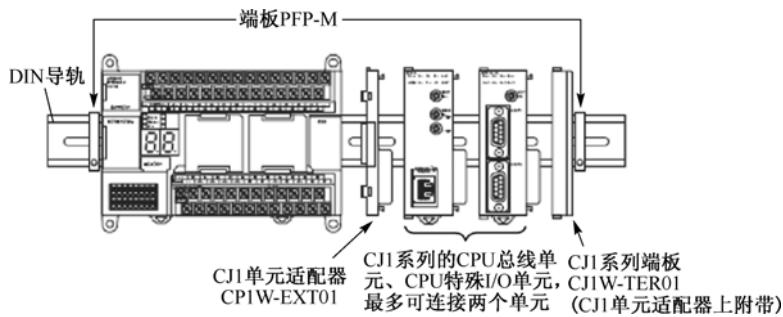


图 2.7 CP1H CPU 单元连接 CJ1 系列高性能单元的配置图

表 2.15 为 CP1 系列可连接的 CJ1 系列的高性能单元。

表 2.15 CP1 系列可连接的 CJ1 系列高性能单元

类 别	单 元 名 称	型 号
CPU 总线单元	Ethernet 单元	CJ1W-ETN11/21
	Controller Link 单元	CJ1W-CLK21-V1
	串行通信单元	CJ1W-SCU21-V1
		CJ1W-SCU31
		CJ1W-SCU41-V1
特殊 I/O 单元	DeviceNet 单元	CJ1W-DRM21
	模拟输入单元	CJ1W-AD041-V1/AD081-V1
	模拟输出单元	CJ1W-DA041/042, CJ1W-DA08V/08C
	模拟输入/输出单元	CJ1W-MAD42
	过程输入单元	CJ1W-PTS51/52
		CJ1W-PTS15/16
		CJ1W-PDC15
	温度调节单元	CJ1W-TC001~004/TC101~104
	高速计数单元	CJ1W-CT021
	ID 传感器单元	CJ1W-V600C11/CJ1W-V600C12
	CompoBus/S 主站单元	CJ1W-SRM21
	位置控制单元	CJ1W-NCF71
	运动控制单元	CJ1W-MCH71

CP1H 为 CJ1 CPU 总线单元分配 400 个通道，范围为 1500~1899CH。每一个单元为 25 CH，根据单元号分配。若单元号为  $N$ ，则占用  $1500+N\times 25\sim 1500+N\times 25+24$ ，即#0、#1、#3、…、#15 单元分别占用 1500~1524 CH、1525~1549 CH、1550~1574 CH、…、1875~1899 CH。关于 CJ1 系列 CPU 总线单元分配的通道内容，请参见各单元的用户手册。

CP1H 为 CJ1 特殊 I/O 单元分配 960 个通道，范围为 2000~2959 CH，每一个单元为 10 CH，根据单元号分配。若单元号为  $N$ ，则占用  $2000+N\times 10\sim 2000+N\times 10+9$ ，即#0、#1、#3、…、#95 单元分别占用 2000~2009 CH、2010~2019 CH、2020~2029 CH、…、2950~2959 CH。关于 CJ1 系列特殊 I/O 单元分配的通道内容，请参见各单元的用户手册。

2.3 CP1 系列 PLC 的存储器分配

CP1 系列 PLC 的内部继电器和数据区以通道形式进行编号，通道号用 3~5 位数表示。一个通道内有 16 个继电器，一个继电器对应通道中的一位，16 个位的序号为 00~15。所以一个继电器的编号由两部分组成，一部分是通道号，另一部分是该继电器在通道中的位序号。

CP1 系列 PLC 的存储器分为 5 部分：用户程序存储器、I/O 存储器、参数区、内置闪存和存储盒。其中 I/O 存储器区是指通过指令的操作数可以进入的区域。I/O 存储器区分为通道 I/O (CIO) 区、工作继电器区 (WR)、保持继电器区 (HR)、辅助继电器区 (AR)、暂存继电器区 (TR)、定时器区 (TIM)、计数器区 (CNT)、数据存储区 (DM)、变址寄存器区 (IR)、数据寄存器区 (DR)、任务标志区 (TK) 等。

I/O 存储器区主要是用来存储输入、输出数据和中间变量，提供定时器、计数器、寄存器等，还包括系统程序所使用和管理的系统状态和标志信息。

在对数据区进行操作时，DM 区和 DR 区只能读取字，不能定义其中的某一位。而在 CIO、H、A 和 W 区中可以存取数据的字或位，这取决于操作数的指令。

2.3.1 CP1H/CP1L 系列 PLC 的 I/O 存储器分配

CP1H I/O 存储器的分配见表 2.16。CP1L 和 CP1H 的内部存储器分配类似，CP1L CPU 单元点数种类丰富，没有内置模拟量功能和不能连接 CJ 系列特殊 I/O 单元，与 CP1H 不同之处标注于表 2.16 中。

表 2.16 CP1H-XA 型的 I/O 存储器的分配

区 域			大小 (范围)	注 释
通道 I/O 区 (CIO)	开关量输入/输出继电器 (CP1L 类似, 详情见手册)	内 置 开 关 量 输入/输出继电器	24 个输入点 (0.00~0.11, 1.00~1.11)	内置输入继电器为 CPU 主机单元带有的内置输入继电器。可直接对外输入
			16 个 输 出 点 (100.00 ~ 100.07, 101.00~101.07)	内置输出继电器为 CPU 主机单元带有的内置输出继电器。可直接对外输出
		扩 展 开 关 量 输入/输出继电器	15 CH (2 CH~16 CH)	扩展输入继电器区, 可直接对外输入
			15 CH (102 CH~116 CH)	扩展输出继电器区, 可直接对外输出
	内置模拟输入/输出继电器 (限 XA 型)		4 CH (200~203 CH)	内置模拟输入继电器, 可直接对外输入 (CP1L 无)
			2CH (210~211 CH)	内置模拟输出继电器, 可直接对外输出 (CP1L 无)
	数据链接继电器		3200 位, 200 CH (1000 CH~1199 CH)	用于 Controller Link 的数据链接 (CP1L 为 3000 CH~3063 CH)
	CJ 系列 CPU 总线单元继电器		6400 位, 400 CH (1500 CH~1899 CH)	连接 CJ 系列 CPU 总线单元时使用, 每单元 25CH, 最多 16 单元 (CP1L 无)
	CJ 系列特殊 I/O 单元继电器		15360 位, 960 CH (2000 CH~2959 CH)	连接 CJ 系列特殊 I/O 单元时使用, 每单元 10CH, 最多 96 单元 (CP1L 无)

(续表)

区 域		大小(范围)	注 释
通道 I/O 区 (CIO)	串行 PLC 链接继电器	1440 位, 90 CH (3100 CH~3189 CH)	串行 PLC 链接中使用的区域, 用于与其他 PLC CP1H CPU 单元或 CJ1M CPU 单元进行的数据链接
	DeviceNet 继电器	9600 位, 600 CH (3200 CH~3799 CH)	使用 CJ 系列 DeviceNet 单元的远程 I/O 主站功能时, 各从站被分配的继电器区域 (CP1L 无)
	内部工作继电器区	4800 位, 300 CH (1200 CH~1499 CH) 37504 位, 2344 CH (3800 CH~6143 CH)	这些位用于编程中, 不能直接对外输入/输出。作为内部辅助继电器优先使用下一行的 W 区 (CP1L 只有 3800 CH~6143 CH)
工作继电器区 (WR)		8192 位, 512 CH (W000 CH~W511CH)	用于编程, 不能直接对外输入/输出 作为内部辅助继电器优先使用该区
保持继电器区 (HR)		8192 位, 512 CH (H000 CH~H511 CH)	保持继电器具有断电保持功能
辅助继电器区 (AR)		只读: 7168 位, 448 CH (A000 CH~A447 CH) 读/写: 8192 位, 512 CH (A448 CH~A959 CH)	系统中被分配特定的功能的继电器。详细内容请参见相关手册
暂存继电器区 (TR)		16 位 (TR00~TR15)	在电路的分支点, 暂时存储 ON/OFF 状态的继电器
定时器区 (TIM)		4096 个 (T0000~T4095)	做定时用
计数器区 (CNT)		4096 个 (C0000~C4095)	做计数用
数据存储器区 (DM)		D00000~D32767 (除右列的用途外, 其他区域作为普通 DM)	D20000~D29599 (100 CH ×96 单元): CJ 系列特殊 I/O 单元用。 D30000~D31599 (100 CH ×16 单元): CJ 系列 CPU 总线单元用 Modbus-RTU 简易主站用固定分配区域: D32200~D32299 (串行端口 1)、D32300~D32399 (串行端口 2)
变址寄存器区 (IR)		16 个, IR0~IR15	储存间接寻址的 PLC 存储器的地址, 一个寄存器 32 位
数据寄存器区 (DR)		16 个, DR0~DR15	储存用于间接寻址的偏移值, 一个寄存器 16 位
任务标志区 (TK)		32 点, TK00~TK31	任务标志是只读标志, 当相应的循环任务在执行时, 则标志为 ON; 当对应任务没有执行或为待机状态时, 标志为 OFF
状态标志区 (CF)		14 位	反映指令执行结果的专用标志, 出错 (ER) 标志、进位 (CY) 标志等 14 个标志, 详见 3.1 节
时钟脉冲区 (CF)		5 位 (P_0_02s, P_0_1s, P_0_2s, P_1s, P_1m)	为系统产生的脉冲, 在 CX-P 软件的全局符号中可查找到

注: CIO 区中不使用的继电器编号可作为内部辅助继电器使用。

1. 通道 I/O 区 (CIO)

通道 I/O 区 (CIO): 其地址前面不必加英文字母符号。例如, 零通道记为 0000CH 或 0000, 而不是 CIO0000。而其他继电器区通道的前面一定要加相应区域的符号。

CIO 区分为 8 部分。

① 内置开关量输入/输出继电器区和扩展开关量输入/输出继电器区 (输入: 0~16 CH, 输出: 100~116 CH): 可以直接对外输入/输出的继电器区域。

内置开关量输入/输出继电器区是 CP1H CPU 主机单元固有的输入/输出点, 共 40 点。

24 个输入点, 占输入的两个通道: 0.00~0.11, 1.00~1.11。

16 个输出点, 占输出的两个通道: 100.00~100.07, 101.00~101.07。

扩展开关量输入/输出继电器区: 当 CP1H CPU 主机单元连接 CP1W 或 CPM1A 扩展单元时, 扩展单元所占的通道号。

扩展输入继电器, 可占用的输入通道: 15 CH (2~16 CH)。

扩展输出继电器, 可占用的输出通道: 15 CH (102~116 CH)。

不使用的继电器编号可作为内部辅助继电器使用。

② 内置模拟量输入/输出继电器区 (仅限 XA 型), 用于分配 CP1H CPU 单元 XA 型的内置模拟输入/输出的继电器区域。内置模拟量输入占用 4CH: 200~203 CH, 输出占用 2CH: 210~211 CH。

③ 数据链接继电器, 占用 3200 位 (200 CH): 1000~1199 CH。用于 Controller Link 中的数据链接, 或 PLC 链接系统中的 PC 链接。数据链接是指通过安装在各 PLC 上的 Controller Link 单元所构成的网络, 自动地访问网络中其他 PLC, 实现链接区的数据共享。数据链接可以自动创建或人工创建。详细内容参见 Controller Link 单元操作手册。不使用的继电器编号可作为内部辅助继电器使用。

④ CJ 系列 CPU 总线单元继电器, 占用 6400 位 (400 CH): 1500~1899 CH。连接 CJ 系列 CPU 总线单元时使用, 每单元 25CH, 最多 16 单元。某单元占用的通道号:  $1500 + \text{单元号} \times 25 \sim 1500 + \text{单元号} \times 25 + 24$ , 即 #0、#1、#3、…、#15 单元分别占用 1500~1524 CH、1525~1549 CH、1550~1574 CH、…、1875~1899 CH。

不使用的继电器编号可作为内部辅助继电器使用。

⑤ CJ 系列特殊 I/O 单元继电器, 占用 15 360 位 (960 CH): 2000~2959 CH。

连接 CJ 系列特殊 I/O 单元时使用, 用于传送单元操作状态等数据。每单元分配 10 字, 最多 96 单元。某单元占用的通道号:  $2000 + \text{单元号} \times 10 \sim 1500 + \text{单元号} \times 10 + 9$ , 即 #0、#1、#3、…、#95 单元分别占用 2000~2009 CH、2010~2019 CH、2020~2029 CH、…2950~2959 CH。

不使用的继电器编号可作为内部辅助继电器使用。

⑥ 串行 PLC 链接继电器, 占用 1440 位 (90CH): 3100~3189CH, 是串行 PLC 链接中使用的区域, 两个相同或不同 PLC 之间的数据链接, 例如, CP1H 之间或 CP1H 与 CJ1M 之间的数据链接。串行 PLC 链接通过 RS-232C 端口, 进行 CPU 单元之间的数据交换。串行 PLC 链接区的通道分配需根据主站中的 PLC 系统设定而自动设定。不使用的继电器编号可作为内部辅助继电器使用。

⑦ DeviceNet 继电器, 占用 9600 位 (600CH): 3200~3799CH。

使用 CJ 系列 DeviceNet 单元的远程 I/O 主站功能时, 各从站被分配的继电器区域。不使用时, 该区域可作为内部辅助继电器使用。

⑧ 内部工作继电器区。CIO 中的内部工作继电器区占用两部分: 4800 位 (300 CH): 1200~1499 CH; 37504 位 (2344 CH): 3800~6143 CH。

仅可在程序上使用的继电器区域, 不可以直接对外输入/输出。工作继电器区有两部分, 相比该区域, 优先使用下面的 WR 区域。因为该区域可能根据将来 CPU 单元的版本升级被分配特定的功能。

## 2. 工作继电器区 (WR)

占用 8192 位 (512 CH): W000~W511 CH。

工作继电器区是指不可以直接对外输入/输出的继电器区域。这些字只能在程序内使用, 它们不能用于与外部 I/O 端子的 I/O 信息交换, 可作为程序中的中间继电器使用。

## 3. 保持继电器区 (HR)

占用 8192 位 (512 CH): H000~H511 CH。

保持继电器用于存储/操作各种数据并可按字或按位存取, 在字号前要冠以“H”字符, 以区别于其他区。

当系统操作方式改变、电源中断或 PLC 操作停止时, 保证继电器能够保持其状态。

H512~H1535 CH 为功能块专用保持继电器。仅可在功能块 FB 实例区域 (变量的内部分配范围) 设定。

#### 4. 辅助继电器区 (AR)

占用 15 360 位 (960 CH): A000~A959。

辅助继电器区用来存储 PLC 的工作状态信息,如特殊 I/O 单元的错误标志、链接系统操作错误标志、远程 I/O 主单元错误标志、从站机架错误标志、特殊 I/O 单元重启动、链接系统操作重启动、远程 I/O 单元重启动、时钟设置位及数据跟踪标志等。

辅助继电器包括系统自动设定的继电器和用户进行设定操作的继电器。由自诊断发现的异常标志、初始设定标志、操作标志、运行状态监视数据等构成。

#### 5. 暂存继电器区 (TR)

占用 16 位: TR00~TR15。

在电路的分支点,暂时存储 ON/OFF 状态的继电器。关于使用方法,可参见 3.4.3 节。

#### 6. 计数器 (CNT)/定时器 (TIM) 区

CP1H 有定时器 4096 个: T0000~T4095,有计数器 4096 个: C0000~C4095。

定时器用于需要定时、延时 ON 及延时 OFF 等场合。计数器用于记录外部输入脉冲信号,计数器分为两种,一种是单向计数器,另一种是双向计数器,也称可逆计数器。双向计数器有两个脉冲输入端,当“加”脉冲输入端有一个脉冲输入时,计数器的存储器内容加 1,当“减”脉冲输入端有一个脉冲输入时,计数器的存储器内容减 1。

#### 7. 数据存储器区 (DM)

数据存储器区是一个只能以字为单位存取的多用途数据区。数据存储器 (DM) 用于内部数据的存储和处理,如数据传送、数值运算、数据转换、数值比较的结果,逻辑指令、特殊指令、网络指令、串行通信指令、模拟量输入单元、模拟量输出单元、高速计数单元及定位控制单元的参数设定、处理结果等。数据存储器区只能进行字操作,不能用于位操作。

欧姆龙公司的 CP1H 系列将数据存储器分为 4 个区:普通 DM、CJ 系列特殊 I/O 单元用区、CJ 系列 CPU 总线单元用区和 Modbus-RTU 简易主站用区。

CJ 系列特殊 I/O 单元占用: D20000~D29599 (100CH × 96 单元)。

CJ 系列 CPU 总线单元占用: D30000~D31599 (100 CH × 16 单元)。

Modbus-RTU 简易主站固定分配区域:

D32200~D32299 (串行端口 1)、D32300~D32399 (串行端口 2)。

普通 DM: D00000~D32767 中,除上面已用的区域。

#### 8. 变址寄存器区 (IR)

变址寄存器有 16 个: IR0~IR15,1 个寄存器 32 位。用于间接寻址一个字,每个变址寄存器存储一个 PLC 存储地址,该地址是在 I/O 存储区中一个字的绝对地址。

#### 9. 数据寄存器区 (DR)

数据寄存器有 16 个: DR0~DR15,1 个寄存器 16 位。储存用于间接寻址的偏移值。

间接寻址中利用 16 个数据寄存器 (DR0~DR15) 来偏移变址寄存器的 PLC 存储地址。将数据寄存器中的值加到变址寄存器的 PLC 存储地址上,来指定一个位或字在 I/O 存储区中的绝对地址,数据寄存器中的数据取值范围为-32768~32767,偏移量的范围由此确定。

数据寄存器只能进行字操作,不能用于位操作。



10. 任务标志区(TK)

任务标志有 32 个：TK00～TK31。

任务标志是只读标志，当相应的循环任务在执行时，则标志为 ON；当对应任务没有执行或为待机状态时，标志为 OFF。

11. 状态标志/时钟脉冲区(CF)

状态标志/时钟脉冲的地址都是以 CF 开始。

状态标志是根据指令的执行结果更新的标志，详见 3.1 节。

时钟脉冲是由系统产生的，有 5 个时钟脉冲，分别为 P\_0\_02s(0.02s)、P\_0\_1s(0.1s)、P\_0\_2s(0.2s)、P\_1s(1s)和 P\_1m(1min)，可以用于编程。

2.3.2 CP1E 系列 PLC 的 I/O 存储器分配

CP1E 的特点是简单和经济。CP1E 与 CP1H 的内部存储器分配及容量差别较大，见表 2.17。

表 2.17 CP1E 的 I/O 存储器区的分配

区 域		大小(范围)	注 释
通道 I/O 区 (CIO)	开关量输入/ 输出继电器	100 CH(0CH～99 CH)	内置和扩展输入继电器区，可直接对外输入
		100 CH(100 CH～199CH)	内置和扩展输出继电器区，可直接对外输出
	内置模拟量 (限 NA 型)	输入：2 CH(90～91CH)	内置模拟输入继电器，可直接对外输入
		输出：1CH(190CH)	内置模拟输出继电器，可直接对外输出
	串行 PLC 链 接继电器	1440 位，90 CH (200 CH～289 CH)	串行 PLC 链接中使用的区域，用于与其他 CP1 PLC 连接
工作区继电器(WR)		1600 位，100 CH(W00 CH～W99CH)	作为内部辅助继电器用于编程，不能直接对外输入/输出；
保持继电器 (HR)		800 位，50CH(H00CH～H49 CH)	保持继电器具有断电保持功能
辅助继电器(AR)		只读：7168 位，448 CH(A000 CH～A447 CH) 读/写：4896 位，306 CH(A448 CH～A753 CH)	系统中被分配特定的功能的继电器。详细内容请参见相关手册
暂存继电器(TR)		16 位(TR00～TR15)	在电路的分支点，暂时存储 ON/OFF 状态的继电器
定时器(TIM)		256 个(T000～T255)	作定时用
计数器(CNT)		256 个(C000～C255)	作计数用
数据存储器(DM)		E 型 CPU 单元 2 K 字 D0～D2047	Modbus-RTU 简易主站用固定分配区域:D1200～D1299 (串行端口 1)、D1300～D1399(串行端口 2)
		N/NA 型 CPU 单元 8 K 字 D0～D8191	E 型：D0～D1499 可备份到备份存储器(内置 EEPROM) N/NA 型：D0～D6999 可备份到备份存储器
状态标志(CF 区)		14 位	反映指令执行结果的专用标志，出错(ER)标志、进位(CY)标志等 14 个标志，详见 3.1 节 3 段内容
时钟脉冲(CF 区)		5 位(P_0_02s, P_0_1s, P_0_2s, P_1s, P_1m)	为系统产生的脉冲，在 CX-P 软件的全局符号中可查找到

注：CIO 区中不使用的继电器编号可作为内部辅助继电器使用。

CP1H、CJ1 和 CS1 的内部存储器设置趋向一致，存储器软器件的种类几乎是一样的。当然，同一种内部器件里是会有所不同的。例如，因为功能的不同，两种 PLC 的 CIO 区划分有差异；同一种内部器件，在数量上，一般 CS1 要多于 CP1H 和 CJ1。

另外，CJ1、CS1 系列 PLC 的存储器比 CP1H 多了扩展数据存储器(EM)。EM 区是 DM 区的扩充，也具有断电保持功能。CJ1 的 EM 区分为若干段(Bank)，每个段 32 767 个字，最多 7 个段(由 PLC 型号决定)，范围为 E0\_00000～E6\_32767。

## 2.4 CJ1/CS1 的 I/O 通道分配

### 2.4.1 CJ1 的 I/O 通道分配

CJ1/CS1 系列 PLC 属组合式(模块式)结构。

CJ1 的 CPU 机架由供电单元、CPU 单元、基本 I/O 单元、特殊 I/O 单元、CPU 总线单元和一个终端板组成。如果连接扩展机架，要安装一个 I/O 控制单元。CJ1 的扩展机架由供电单元、I/O 接口单元、基本 I/O 单元、特殊 I/O 单元、CPU 总线单元和一个终端板组成。

CJ1 的 CPU 机架和扩展机架上安装基本 I/O 单元、特殊 I/O 单元、CPU 总线单元，每个机架上安装的单元数目不能超过 10 个。

如图2.8所示，CJ1 在 CPU 机架上安装一个 I/O 控制单元，在每个扩展机架上安装一个 I/O 接口单元，最多允许连接 3 个扩展机架。这就能组成一个包含多达 40 个单元的机架的系统，即 10 个单元的 CPU 机架和 30 个单元(10×3)的扩展机架。

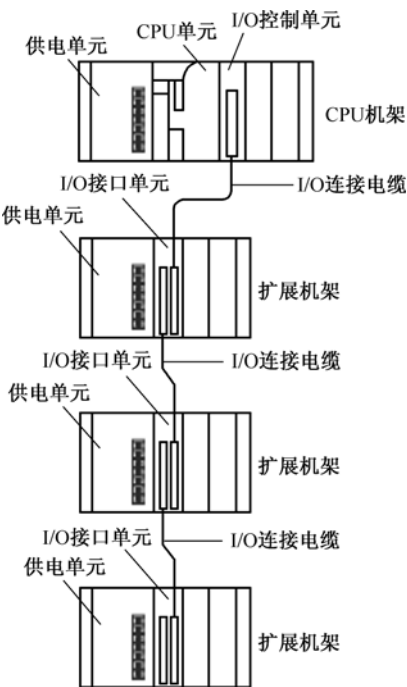


图 2.8 CJ1 的扩展配置

#### 1. CJ1 基本 I/O 单元的通道分配

CJ1 的 I/O 继电器编号为 6 位数。前 4 位为通道号，后两位为点号。点在模块中的编号是固定的，占两位数，与模块上标明的点号对应。16 点或少于 16 点的模块占用 1 个通道，多于 16 点但不超过 32 点的模块占用两个通道，多于 32 但不超过 64 点的模块占用 4 个通道。

基本 I/O 单元地址通道(字)的范围是 CIO0000~CIO0159。

基本 I/O 单元通道由模块所在机架的位置及模块在机架上的位置决定。地址通道的分配顺序为先 CPU 机架，后扩展机架；先靠近 CPU 机架的扩展机架，后远离 CPU 机架的扩展机架；在一个机架内，先左后右(先靠近电源的，后远离电源的)，按顺序增加编号。

默认情况下，前一个机架的最后一个模块的编号与后一个机架的第一个模块的编号是顺序衔接的。这种情况也可以改变，机架上的第一字可用编程工具设定为 CIO0000~CIO0159 之间的任意字，然后其他模块依次编号。但不管怎样做，每一模块的编号是唯一的，且不出 CIO0000~CIO0159 的范围。

图2.9为 CPU 机架通道的分配，其中图2.9(a)为分配顺序，图2.9(b)为分配实例，机架最左边的第一个字用默认的 CIO0000。图2.9(b)中有两个输入模块，一个为 16 点，另一个为 32 点。3 个输出模块中，16 点、32 点、64 点各占一个。

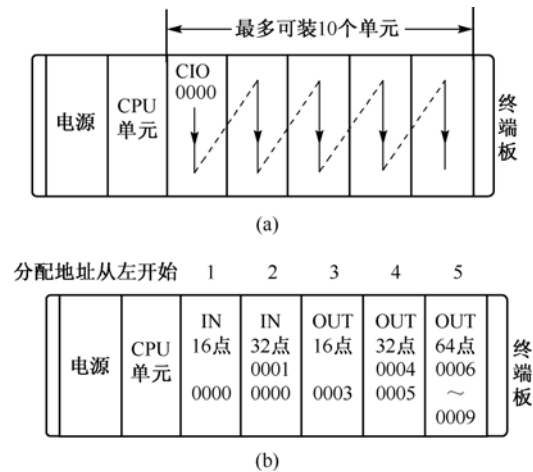


图 2.9 CPU 机架通道的分配

各个模块均已分配了通道号。所有机架的第 1 个模块按默认方式分配通道号，CPU 机架的第一个模块

通道号为 0000，最末 1 个模块的通道号排到 0009，所以，第 1 个扩展机架的第 1 个模块的通道号为 0010，第 1 个扩展机架的最末 1 个模块的通道号排到 0015，所以，第 2 个扩展机架的第 1 个模块的通道号为 0016。

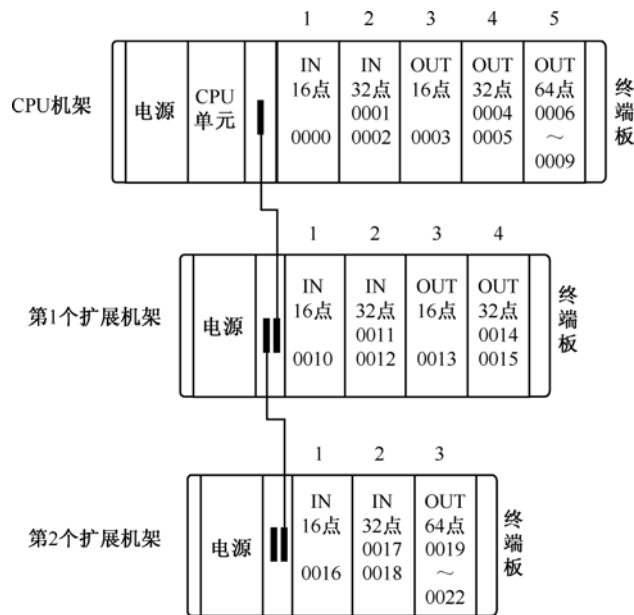


图 2.10 多机架通道的分配实例

在基本 I/O 单元的 I/O 通道分配期间，特殊 I/O 单元和 CPU 总线单元将被忽略，即它们的插槽被视为空槽。对这两类单元用其他办法为它们分配通道号。

2. CJ1 特殊 I/O 单元的通道分配

CJ1 的特殊 I/O 单元有自己的单元号，由其上的设定开关设定，设定的范围为 00~95。这并不意味着 CJ1 可以控制 96 个特殊 I/O 单元，因为 CJ1 最大的可能槽数为 40，最多安装 40 个单元。

CJ1 的特殊 I/O 单元占用 I/O 通道 CIO2000~CIO2959 和数据存储区 DM20000~CIO29599。每个单元根据单元号分配 10 个 I/O 通道和 100 个 DM 字。但有的特殊 I/O 单元要占用两个单元号甚至更多，例如，位控单元占用两个单元号，即占用 20 个 I/O 通道和 200 个 DM 字，下一个单元号设置时要隔开。设置单元号时要注意避免重复。

若单元号为  $N$ ，则占用 I/O 通道  $2000+N\times10\sim2000+N\times10+9$ ，即#0、#1、#3、…、#95 号单元分别占用 CIO2000~CIO2009、CIO2010~CIO2019、CIO2020~CIO2029、…、CIO2950~CIO2959。占用 DM 字  $20000+N\times100\sim20000+N\times100+99$ ，即#0、#1、#3、…、#95 号单元分别占用 DM20000~DM20099、DM20100~DM20199、DM20200~DM20299、…、DM29500~DM29599。

图2.11 为 CJ1 特殊 I/O 单元的通道分配，CPU 机架上有 6 个模块，其中，两个为特殊 I/O 单元，单元号分别设为#00、#01。

3. CJ1 CPU 总线单元的通道分配

CJ1 的 CPU 总线单元也有自己的单元号，由其上的设定开关设定，设定的范围为 00~16。这意味着 CJ1 最多可以控制 16 个 CPU 总线单元。



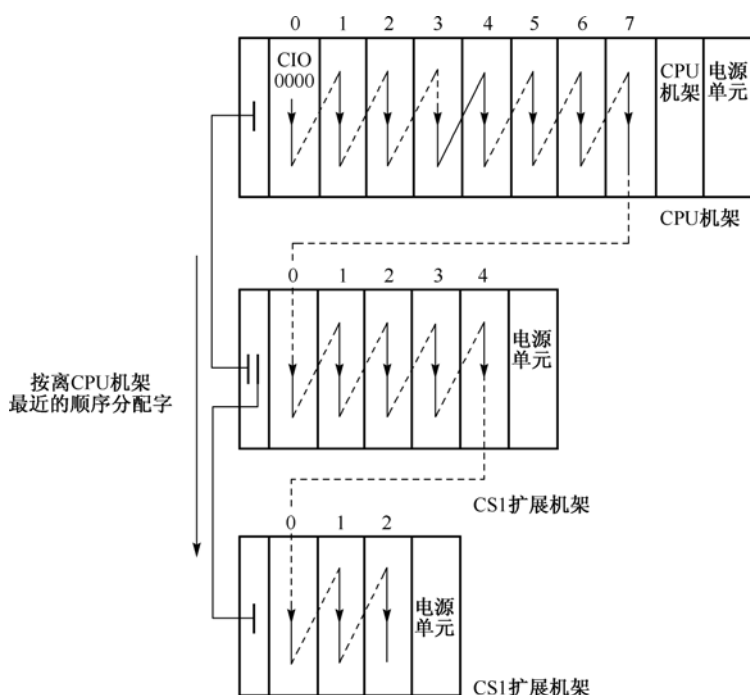


图 2.13 CS1 I/O 单元通道分配顺序

若单元号为  $N$ ，则占用 I/O 通道  $2000+N \times 10 \sim 2000+N \times 10+9$ ，即#0、#1、#3、…、#95 号单元分别占用 CIO2000~CIO2009、CIO2010~CIO2019、CIO2020~CIO2029、…、CIO2950~CIO2959。占用 DM 字  $20000+N \times 100 \sim 20000+N \times 100+99$ ，即#0、#1、#3、…、#95 号单元分别占用 DM20000~DM20099、DM20100~DM20199、DM20200~DM20299、…、DM29500~DM29599。

### 3. CPU 总线单元的通道分配

CS1 的 CPU 总线单元也有自己的单元号，由其上的设定开关设定，设定的范围为 00~16。这意味着 CS1 最多可以控制 16 个 CPU 总线单元。

CS1 的 CPU 总线单元占用 I/O 通道 CIO1500~CIO1899 和数据存储区 DM30000~DM31599。每个单元根据单元号分配 25 个 I/O 通道和 100 个 DM 字。设置单元号时，要注意避免重复。

若单元号为  $N$ ，则占用 I/O 通道  $1500+N \times 25 \sim 1500+N \times 10+24$ ，即#0、#1、#3、…、#16 号单元分别占用 CIO1500~CIO1524、CIO1525~CIO1549、CIO1550~CIO1574、…、CIO1875~CIO1899。占用 DM 字  $30000+N \times 100 \sim 30000+N \times 100+99$ ，即#0、#1、#3、…、#95 号单元分别占用 DM30000~DM30099、DM30100~DM30199、DM30200~DM30299、…、DM39500~DM39599。

## 习 题

1. CP1H 系列 PLC 属于何种结构的 PLC？CP1H 的突出特点是什么？
2. X 型、XA 型与 Y 型 CP1H 的性能各有何特点？
3. 叙述 CP1H 主机面板的各端子和端口的作用。
4. CP1H 系列 PLC 最多可扩展成多少个 I/O 点的系统？其 I/O 点如何编号？
5. CP1H 系列 PLC 扩展方式是什么？分别如何与 CPM1W、CPM1A 系列扩展单元和 CJ 系列高性能单元连接？

6. CP1H 系列 PLC 的编程工具有哪几种? 编程工具与 PLC 怎样连接?
7. CP1H 系列 PLC 的内部继电器区是怎样划分的?
8. XA 型 CP1H 如何设定模拟量输入信号的型式(电压型或电流型)?
9. CP1H CPU 单元的模拟设定电位器与外部模拟量输入连接器有什么区别? 设定的数据存在何处?
10. 快速响应输入功能有何作用? 哪些输入点是快速响应输入点? 其可接收的最窄输入脉冲宽度为多少?
11. CP1 系列 PLC 包括几种类型? 说说这几类 PLC 的异同点?
12. 中断有什么作用? CP1H 系列 PLC 的中断输入点有哪些?
13. CP1H 系列 PLC 有几个高速计数器? CP1HXA 型 PLC 最高计数频率是多少?
14. CP1H 系列 PLC 有几路高速脉冲输出? 其最高输出脉冲频率是多少?
15. 为什么继电器型的 PLC 不宜输出高速脉冲信号?
16. CP1H 系列 PLC 的输出单元有哪些类型? 它们各有什么特点? 适合哪些场合?
17. 一台 CP1H, 加两个 20 点 CPM1W 扩展单元, 如何分配通道和地址?
18. CJ1 系列 PLC 的 CPU 机架由哪些单元组成? 扩展机架又由哪些单元组成? 每个机架上安装的单元数目最多几个? 画出有 CPU 机架和扩展机架的 CJ1 系列 PLC 的组成图。
19. CJ1 系列 PLC 特殊 I/O 单元的通道如何分配? CPU 总线单元的通道又如何分配?
20. CS1 系列 PLC 特殊 I/O 单元的通道如何分配? CPU 总线单元的通道又如何分配?
21. CP1H、CP1L 和 CP1E PLC 最多能扩展的点数各为多少? 分别需要采用什么型号的 CP1H、CP1L 和 CP1E 连接什么类型的扩展单元来实现? 分别画出连接图, 并标明 I/O 点的编号。

# 第 3 章 CP1 系列 PLC 的指令系统

PLC 是通过程序对系统实现控制的，因此一种机型的指令系统在一定程度上反映出其控制功能的强弱。CPM1A 和 CPM2A 是世纪之交流行的 OMRON 小型机，其指令约为 100 条。CP1 系列 PLC 也属于小型机，但它的指令系统却非常丰富，约 500 种。CP1H 系列 PLC 运行速度快，它的基本指令(如 LD 指令)执行时间为 0.1  $\mu$ s/条，应用指令(如 MOV 指令)的执行时间为 0.3  $\mu$ s/条，CP1L 执行时间分别为 0.55  $\mu$ s/条和 4.1  $\mu$ s/条，CP1E 执行时间分别为 1.19  $\mu$ s/条和 7.9  $\mu$ s/条。而早期的 CPM1A 和 CPM2A 指令的执行时间分别为 1.72  $\mu$ s/条和 16.3  $\mu$ s/条，及 0.64  $\mu$ s/条和 7.8  $\mu$ s/条。

本章以 CP1H 为例介绍 CP1 系列 PLC 指令的功能及使用方法。在理解指令的含义、熟练其使用方法以后，恰当地使用指令，可以发挥它强大的控制功能。

本章的指令是按 CX-P 软件中“查找指令”中指令的目录分类的。最常用的基本指令在 CX-P 的菜单上有快捷按钮，不用查找。

## 3.1 指令概述

CP1 系列 PLC 指令的基本表示方式是梯形图和语句表。从介绍指令开始，在以下各节中将逐一介绍梯形图和语句表编程。

### 1. 指令的分类

按指令功能的不同，可分为基本指令和应用指令两类。基本指令是直接对输入和输出点进行操作的指令，如输入、输出及逻辑“与”、“或”、“非”等操作。应用指令是进行数据传送、数据处理、数据运算、程序控制等操作的指令。应用指令的多少关系到 PLC 功能的强弱。

### 2. 指令的格式

指令的格式可以表示为：

助记符（指令码）操作数 1  
                                操作数 2  
                                操作数 3

① 助记符表示指令的功能，它指明了执行该指令所完成的操作。助记符常用英文或其缩写来表示。对不同生产厂家的 PLC，相同功能的指令其助记符可能不同。

② 指令码是指令的代码，用 3 位十进制数(000~999)表示。大部分基本指令没有指令码，而应用指令几乎都有指令码。

③ 操作数提供了指令执行的对象或数据。各种指令的操作数个数不同，有的指令不带操作数，有的指令带 1 个操作数，有的指令带两个或三个操作数。关于操作数做如下说明：

- 操作数可为继电器号、通道号或常数。为了区别一个操作数是常数还是通道号，在作为操作数的常数前要加前缀“#”或“&”。

例如，计数器指令可表示为

CNT  
N  
SV

其中，N 是计数器的编号，SV 是计数器设定值。若 N=0000，SV=200 时，表明 0000 号计数器的设定值是 CIO 区 200 通道中的数据；若 SV=#200 时，表明计数器的设定值是 BCD 数据 200。

- BCD 码是指十进制的每一位用 4 位二进制表示，其最常用的是 8421 BCD 码。二进制数常被称为 BIN 数据；而十六进制数常用数字后跟 Hex 来表示，本书中把 Hex 简记为 H。
- 操作数为常数时，可以是十进制或十六进制，这取决于指令的要求。常数在 CX-P 中的输入方式有：#+16 进制数表示十六进制数；#+10 进制数表示十进制数；&+10 进制数表示十进制数。十进制数；十六进制数等几种输入方式，要看指令的具体规定。
- 间接寻址的操作数用\*/@Dxxxxx表示。这种操作数是以 DM 中的数据为地址的另一个 DM 通道中的数据。\*Dxxxxx中的内容是 BCD 码，而@Dxxxxx中的内容是 BIN 码。

3. 执行指令对状态标志位的影响

状态标志位是指根据指令执行结果更新的标志。有的指令执行后不影响标志位，有的指令执行后可能影响标志位。在下面介绍每一条指令时将说明其执行后是否影响标志位、影响哪些标志位，并指出可能使这些标志位置位的原因。其中 ER、EQ 和 N 是最常用的标志位，若 ER 为 ON，表示当前执行的程序出错且停止执行程序。

执行指令将影响的状态标志位的名称地址分配见表 3.1。

表 3.1 状态标志位的名称地址分配

名 称	变 量 名	地 址
出错标志	P_ER	CF003
存取出错标志	P_AER	CF011
进位标志	P_CY	CF004
>标志	P_GT	CF005
<标志	P_LT	CF007
=标志	P_EQ	CF006
负标志	P_N	CF008
上溢出标志	P_OF	CF009
下溢出标志	P_UF	CF010
≥标志	P_GE	CF000
≠标志	P_NE	CF001
≤标志	P_LE	CF002
常 ON 标志	P_On	CF113
常 OFF 标志	P_Off	CF114

从表 3.1 中可见，所有的标志位都是以 P\_开始，常常将标志位前面的 P\_省略，如 P\_ER 记为 ER。表中的地址无须记忆，在 CX-P 编程软件中选中标志位变量名，则相应的地址就会显示出来。

4. 指令的执行条件

一般线圈或指令都有执行条件，线圈或指令不能直接与左侧母线连接，必须与继电器触点相连。当继电器触点闭合时，满足执行条件，接通线圈或指令。极少数指令没有执行条件，如 END 等。不能直接与左母线连接的指令，如果不需执行条件时，可以通过特殊辅助继电器 P\_On(常 ON)的触点连接，如图 3.1 所示。

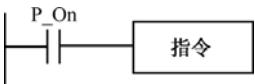


图 3.1 使用常 ON 标志的例子



3.2 基本指令及编程

3.2.1 基本指令

编写应用程序时，使用频率最高的是基本指令。CPIH 系列 PLC 有 10 条基本指令，初学者要从学习这些简单的指令入手，逐步了解其他各种指令的功能和使用方法。

1. 读 LD、读非 LD NOT、与 AND、与非 AND NOT、或 OR、或非 OR NOT、输出 OUT、输出非 OUT NOT 指令

上述基本指令，几乎所有的程序都必须使用。表 3.2 列出了这些常用的基本指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.2 常用的基本指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
LD N 读		范围：CIO、W、H、A、T/C、TK 或 TR 等 以位为单位进行操作	常开触点与左侧母线相连接的指令 指令执行结果不影响标志位
LD NOT N 读非		范围：CIO、W、H、A、T/C 或 TK 等 以位为单位进行操作	常闭触点与左侧母线相连接的指令 指令执行结果不影响标志位
AND N 与		范围：CIO、W、H、A、T/C 或 TK 等 以位为单位进行操作	常开触点与其他程序段相串联的指令 指令执行结果不影响标志位
AND NOT N 与非			常闭触点与其他程序段相串联的指令 指令执行结果不影响标志位
OR N 或			常开触点与其他程序段相并联的指令 指令执行结果不影响标志位
OR NOT N 或非			常闭触点与其他程序段相并联的指令 指令执行结果不影响标志位
OUT N 输出		范围：CIO、W、H、A 或 TR 等 以位为单位进行操作	把运算结果输出到某个继电器的指令 指令执行结果不影响标志位
OUT NOT N 输出非			把运算结果求反输出到某个继电器中 指令执行结果不影响标志位

图3.2使用了部分基本指令，图3.2(a)是梯形图，图3.2(b)是语句表。

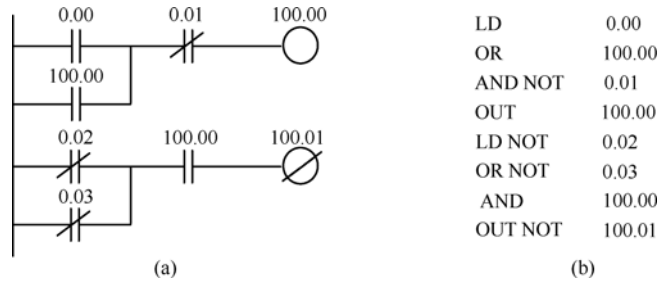


图 3.2 使用基本指令的例子

在分析梯形图程序时，常开和常闭触点的状态(ON/OFF)是由其对应的继电器的状态来确定的。例如，在图3.2中，若 0.00 号输入继电器为 ON，则常开触点 0.00 为 ON(触点闭合)，否则为 OFF；如果 0.01 号输入继电器为 ON，则常闭触点 0.01 为 OFF(触点断开)，否则为 ON。在以后分析程序时，上述原则不再重申。

在图 3.2 中，常开触点 100.00 与 0.00 并联，是逻辑“或”的关系。两者只要有一个为 ON，则并联结果为 ON。常闭触点 0.01 与左面的并联部分相串联，两者是逻辑“与”的关系。常闭触点 0.01 与并联部分的结果都为 ON 时串联结果才为 ON，此结果输出到继电器 100.00 使之成为 ON，否则 100.00 为 OFF。常闭触点 0.02 与常闭触点 0.03 中，只要有一个为 ON、且常开触点 100.00 也为 ON 时，输出继电器 100.01 为 OFF，否则 100.01 为 ON。显然，OUT NOT 指令是把前面结算的结果取反再送到继电器 100.01 中。

2. 块与 AND LD、块或 OR LD 指令

表 3.3 是 AND LD 和 OR LD 指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.3 AND LD 和 OR LD 指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
AND LD 块与		无操作数	并联触点组相串联连接的指令 指令执行结果不影响标志位
OR LD 块或			串联触点组相并联连接的指令 指令执行结果不影响标志位

(1) 块与指令 AND LD

图 3.3 中有 3 个并联的触点组相串联。使用 AND LD 指令时，语句表有如图两种不同的编写方法。方法 2 是把 AND LD 指令集中在一起编写，但方法 2 中 AND LD 指令之前的触点组个数应小于等于 8，而方法 1 对此没有限制。

方法 1

LD	0.00
AND	0.01
OR NOT	0.02
LD	0.03
OR	0.04
AND LD	
LD	0.05
OR NOT	0.06
AND LD	
OUT	100.00

方法 2

LD	0.00
AND	0.01
OR NOT	0.02
LD	0.03
OR	0.04
LD	0.05
OR NOT	0.06
AND LD	
AND LD	
OUT	100.00

图 3.3 使用 AND LD 的例子

(2) 块或指令 OR LD

图 3.4 是 3 个串联的触点组相并联。使用 OR LD 指令时，语句表有如图两种不同的编写方法。同样，在方法 2 中 OR LD 指令之前的触点组个数应小于等于 8，而方法 1 对此没有限制。

方法 1

LD	0.00
AND NOT	0.01
LD NOT	0.02
AND	W0.05
OR LD	
LD	100.04
AND	0.03
OR LD	
OUT	101.00

方法 2

LD	0.00
AND NOT	0.01
LD NOT	0.02
AND	W0.05
LD	100.04
AND	0.03
OR LD	
OR LD	
OUT	101.00

图 3.4 使用 OR LD 指令的示例

### 3.2.2 基本编程规则和编程方法

掌握了 PLC 的基本编程指令之后,就可以根据控制要求编写简单的应用程序了。为了提高编程质量和编程效率,必须首先了解编写梯形图程序的基本编程规则和基本编程方法。

#### 1. 基本编程规则

① 梯形图中的每一行都是从左侧母线开始画起,线圈或指令画在最右边,线圈或指令右边只能画右母线(OMRON PLC 梯形图的右母线可以省略)。

② 线圈或指令不能直接与左侧母线连接(除极少数没有执行条件的指令,如 END 等)。如果必要,可以通过特殊辅助继电器 P\_On(常 ON)的触点与左侧母线连接。

③ 用 OUT 指令输出时,同一编号的继电器线圈在同一程序中使用两次以上,称为双线圈输出。双线圈输出容易引起误动作或逻辑混乱,一般要避免出现这种情况。

例如,图3.5(a)中,设 0.00 为 ON、0.05 为 OFF,由于 PLC 是按扫描方式执行程序的,执行第一行时 100.00 为 ON,而执行第二行时 100.00 为 OFF。在 I/O 刷新阶段 100.00 的输出状态只能是 OFF。显然前面的输出无效,最后一次输出才是有效的。

又如图3.5(b)中,设 0.00 为 ON、0.01 为 OFF。在执行第一行程序后 100.00 为 ON,执行第二行后 100.01 为 ON,执行第三行后 100.00 为 OFF。因此在 I/O 刷新阶段,100.01 为 ON,100.00 为 OFF。但从第二行看,100.00 和 100.01 的状态应该一致。这就是双线圈输出造成的逻辑混乱。

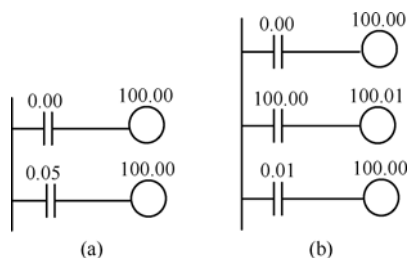


图 3.5 双线圈输出的例子

④ 梯形图必须遵循从左到右、从上到下的顺序编写,不允许在两行之间垂直连接触点。如果不符合上述顺序,就要进行转换。如图 3.6(a)若转换成图 3.6(b)就符合顺序要求了。

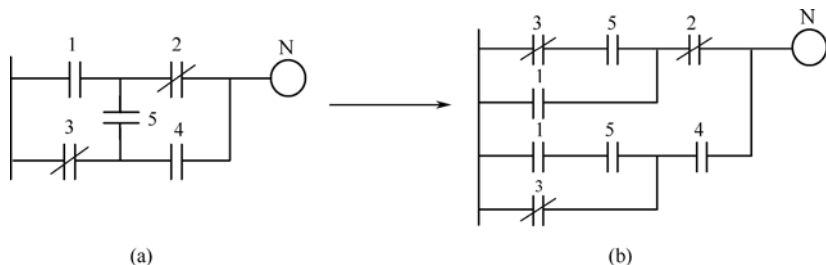


图 3.6 梯形图的顺序转换

⑤ 程序结束时一定要安排 END 指令,否则程序不被执行。本书通常省略 END 指令,以节省空间。

#### 2. 基本编程方法

① 两个或两个以上的线圈或指令可以并联输出,如图3.7所示。

② 触点组与单个触点相并联时,应将单个触点放在下面。如图3.8(a)变成图3.8(b)后,从语句表看出节省了一个 OR LD 语句。

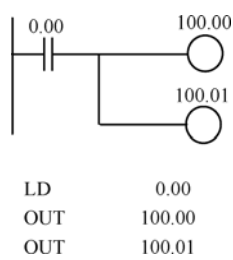


图 3.7 两个线圈并联输出例子

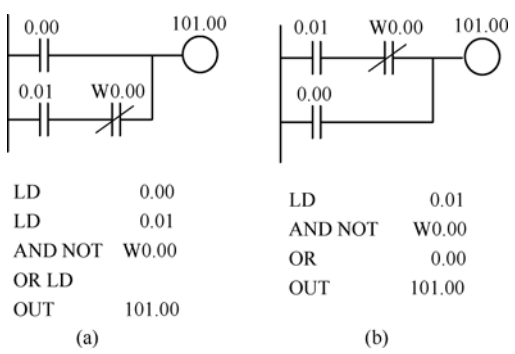


图 3.8 编程方法示例

③ 并联触点组与几个触点相串联时，应将并联触点组放在左边。例如，图3.9(a)变成图3.9(b)后，从语句表看出节省了一个 AND LD 语句。

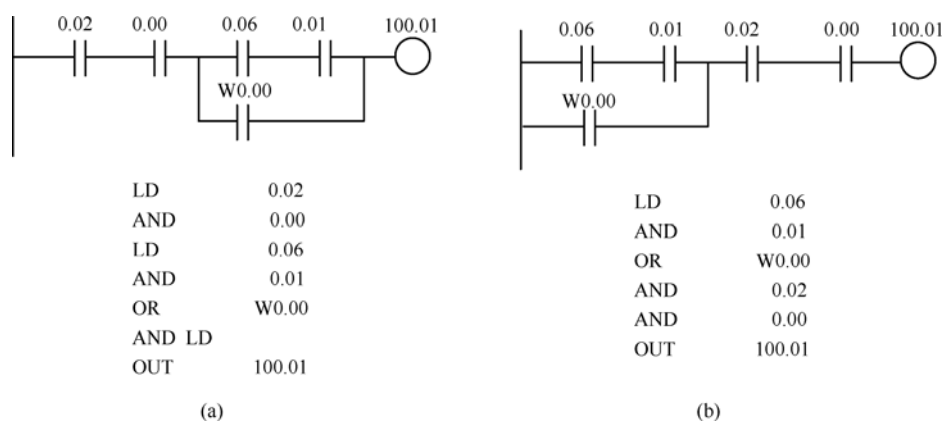


图 3.9 编程方法示例

④ 如果一条指令只需在 PLC 加电之初执行一次，可以用 AR 区的 P\_First\_Cycle (A200.11) 作为其执行条件。由于 P\_First\_Cycle 只在 PLC 加电后的第一个扫描周期处于 ON 状态，因此，P\_First\_Cycle 为执行条件的指令只在加电后的第一个扫描周期被执行。这种用法常出现在 PLC 的初始化程序段上。

图3.10中的 KEEP(保持)指令在后续的 3.3.1 节中将要讲到。在 PLC 加电后的第一个扫描周期，W10.00 被置为 ON，W10.00 又作为 KEEP 指令的置位输入条件，从而使 100.00 被置为 ON。此后，如果复位条件 0.01 ON 使 100.00 复位，则在 PLC 本次加电期间，100.00 不会再被置位。

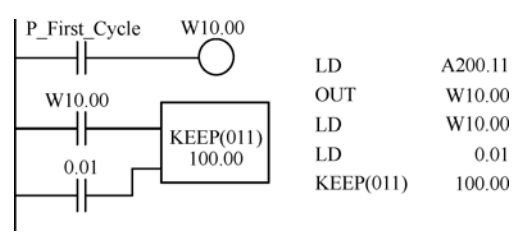


图 3.10 使用 A200.11 的例子

⑤ 有些梯形图难以用 AND LD、OR LD 等基本逻辑指令编写语句表，这时可重新安排梯形图的结构，如图3.11(a)若改画成图3.11(b)就可以使用 OR LD 指令编程了。

⑥ 当某梯级有两个分支时，若其中一条分支从分支点到输出线圈之间无触点，该分支应放在上方，这样可以使语句表的语句更少(详见 3.4.2 节)。

⑦ 尽量使用那些操作数少、执行时间短的指令编程，以缩短扫描周期，从而提高 I/O 响应速度。

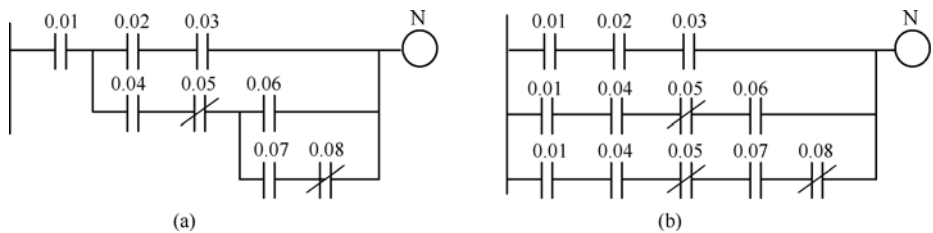


图 3.11 编程方法示例

3.3 顺序输入/输出指令及指令的形式

3.3.1 顺序输入/输出指令

1. 位测试 TST 指令、位测试非 TSTN 指令

表 3.4 列出了位测试指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.4 位测试指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
TST(350) N T 位测试指令		S: 测试数据通道编号 N: 位号#0~F 或&0~15 范围: CIO、W、H、A、T/C、 D、@D、*D 或 DR 等 以位为单位进行操作	TST: 读 S 的 N 位继电器的常开触点 此指令相当于一个常开触点。当指令与左母线相连时, 实现 LD N 功能; 与其他程序段串联时, 实现 AND N 功能; 与其他程序段并联时, 实现 OR N 功能 指令执行结果不影响标志位
TSTN(351) N T 位测试非指令			TSTN: 读 S 的 N 位继电器的常闭触点 此指令相当于一个常闭触点。当指令与左母线相连时, 实现 LD NOT N; 与其他程序段串联时, 实现 AND NOT N; 与其他程序段并联时, 实现 OR NOT N 指令执行结果不影响标志位

位测试(非)指令使用的场合不同时, 分别变成不同类型的位测试(非)指令。与左母线相连时为 LD 型位测试(非)指令, 其功能与 LD (NOT) 指令相同; 与其他程序段串联时为 AND 型位测试(非)指令, 其功能与 AND (NOT) 指令相同; 与其他程序段并联时为 OR 型位测试(非)指令, 其功能与 OR (NOT) 指令相同。可见, 位测试指令与相应的非位测试指令功能一样, 例 LD TST(350) 指令与 LD 指令功能一样。下面分别给出指令的应用示例。

与 LD 指令等价的 TST(350) 指令的应用示例如图 3.12 所示。

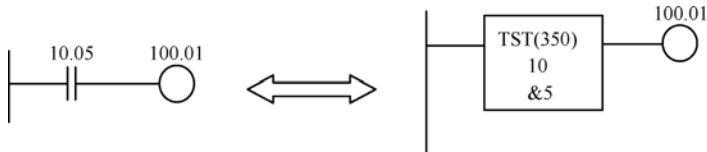


图 3.12 LD 型位测试指令 TST(350) 应用示例

与 LD NOT 指令等价的 TSTN(351) 指令的应用示例如图 3.13 所示。

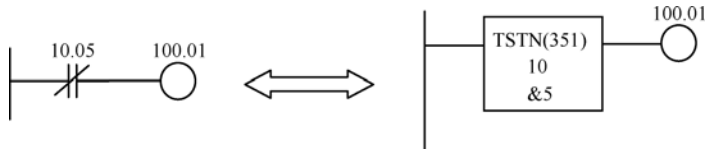


图 3.13 LD 型位测试非指令 TSTN(351) 应用示例

与 AND 指令等价的 TST (350) 指令的应用示例如图3.14所示。

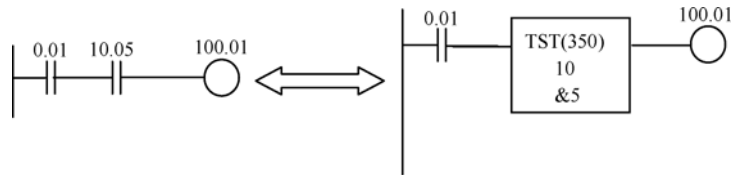


图 3.14 AND 型位测试指令 TST (350) 应用示例

与 AND NOT 指令等价的 TSTN (351) 指令的应用示例如图3.15所示。

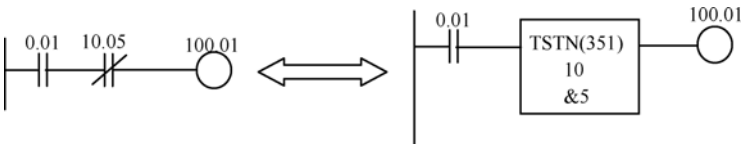


图 3.15 AND 型位测试非指令 TSTN (351) 应用示例

与 OR 指令等价的 TST (350) 指令的应用示例如图3.16所示。

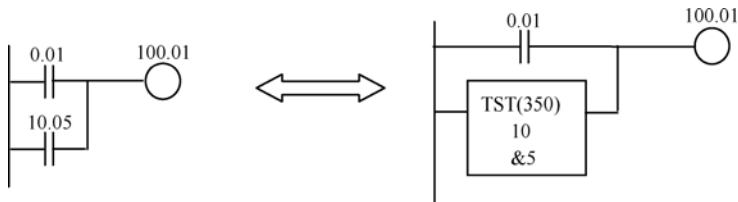


图 3.16 OR 型位测试指令 TST (350) 应用示例

与 OR NOT 指令等价的 TSTN (351) 指令的应用示例如图3.17所示。

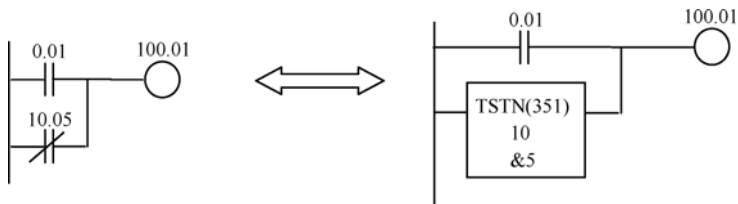


图 3.17 OR 型位测试非指令 TSTN (351) 应用示例

2. 非指令 NOT

将输入条件取反，连接到下一段，如图3.18所示。

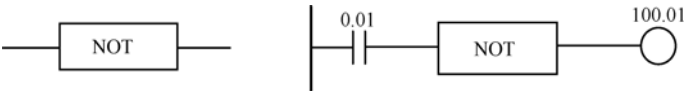


图 3.18 NOT 指令示例

当 0.01 为 OFF，经过 NOT 反向 100.01 为 ON；当 0.01 为 ON，经过 NOT 反向 100.01 为 OFF。

3. 置位 SET 和复位 RSET 指令

表 3.5 是这两个指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.5  SET 和 RSET 指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
SET N 置位指令		范围：CIO、W、H、A 等 以位为单位进行操作	当执行条件为 ON 时，将指定的继电器置为 ON 且保持 指令执行结果不影响标志位
RSET N 复位指令			当执行条件为 ON 时，将指定的继电器置为 OFF 且保持 指令执行结果不影响标志位

SET 和 RSET 指令常成对使用，一般用 SET 指令将某继电器置为 ON，再用 RSET 指令将其置为 OFF。也可以单独用 RSET 指令将已为 ON 的继电器置为 OFF。

SET、RSET 指令的执行条件常使用短信号(脉冲信号)。这两条指令的语句之间可以插入别的指令语句。

图3.19是使用这两个指令的例子，图中，0.00 是 SET 指令的执行条件，当 0.00 为 ON 时，W0.00 被置为 ON 并保持，即使 0.00 又变为 OFF；0.03 是 RSET 指令的执行条件，当 0.03 为 ON 时，W0.00 被置为 OFF 并保持，即使 0.03 又变为 OFF。图3.19(b)是它们的工作波形。当 SET、RSET 指令的操作数是保持继电器 HR 时具有掉电保持功能。

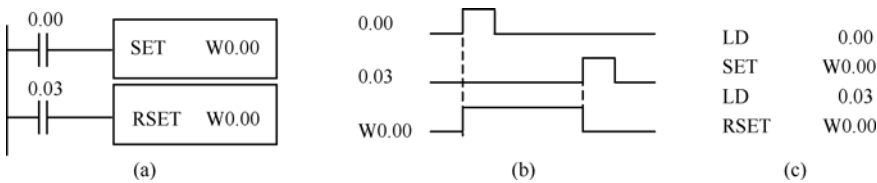


图 3.19  使用 SET 和 RESET 指令的例子

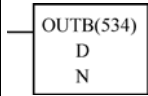
4. 其他置位和复位指令

表 3.6 是其他置位/复位指令的格式/名称、梯形图符号、操作数范围及含义、指令的功能及执行指令对标志位的影响，包括多位和 1 位置位/复位指令。具体用法参见相关手册，本书不再赘述。

表 3.6  其他置位/复位指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
SETA (530) D N1 N2 多位置位指令		D: 置位起始通道号 N1: 置位起始位号, #0~F 或&0~15 N2: 位数#0000~FFFF 或&0~65535 D 范围: CIO、W、H、A449~A959、T/C、D、@D 或*D 等 N1、N2、N 范围: CIO、W、H、A000~A959、T/C、D、@D 或*D、&、DR 等 以位为单位进行操作	当执行条件为 ON 时, 将从 D 通道的 N1 位开始的连续通道的 N2 个连续位置位(置 ON)且保持 对标志位的影响: N1 不在 0000~000F 范围时 ER 为 ON
RSTA (531) D N1 N2 多位复位指令			当执行条件为 ON 时, 将从 D 通道的 N1 位开始的连续通道的 N2 个连续位复位(置 OFF)且保持 对标志位的影响: 同上
SETB (532) D N 1 位置位指令			当执行条件为 ON 时, 将 D 通道的第 N 位置位(置 ON)且保持 对标志位的影响: N 不在 0000~000F 范围时 ER 为 ON
RSTB (533) D N 1 位复位指令			当执行条件为 ON 时, 将 D 通道的 N 位复位(置 OFF)且保持 对标志位的影响: N 不在 0000~000F 范围时 ER 为 ON

(续表)

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
OUTB(534) D N 1 位输出指令		D: 通道号, N: 位号 N: #0~F 或&0~15 D 范围: CIO、W、H、A449~A959、T/C、D、@D 或*D、DR 等 N 范围: CIO、W、H、A000~A959、T/C、D、@D 或*D、&、DR 等 以位为单位进行操作	当执行条件为 ON 时, 将 D 通道的 N 位输出指令执行结果不影响标志位

5. 保持指令 KEEP

表 3.7 是 KEEP 指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.7 KEEP 指令


格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
KEEP(011) N 保持指令	 S 为置位端 R 为复位端 S、R 端可用短信号	范围: CIO、W、H、A 等 以位为单位进行操作	具有锁存继电器的功能 当 S 端输入为 ON 时,继电器 N 被置为 ON 且保持; 当 R 端输入为 ON 时,N 被置为 OFF 且保持;当 S、R 端同时为 ON 时, N 为 OFF N 为 HR 区继电器时有掉电保持功能 指令的执行结果不影响标志位

图 3.20 是使用 KEEP 指令的例子, 请注意用该指令编程时语句表的写法。

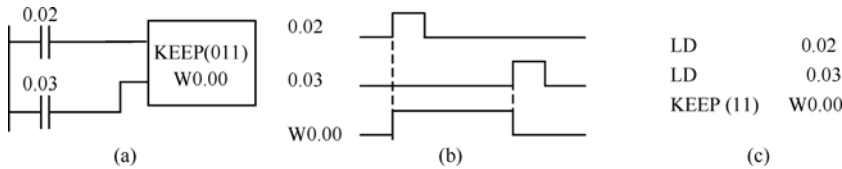


图 3.20 使用 KEEP 指令的例子

在图 3.20 中, 0.02 是置位端的输入条件, 0.03 是复位端的输入条件。当 0.02 由 OFF 变为 ON 时, W0.00 被置为 ON 并保持, 即使 0.02 又变为 OFF; 当 0.03 由 OFF 变为 ON 时, W0.00 被复位为 OFF 并保持, 即使 0.03 又变为 OFF。

比较图 3.19 (b) 和图 3.20 (b) 的波形可看出, 两个程序对 W0.00 都具有启、保、停控制功能。图 3.21 也是实现启、保、停控制的程序。三个图的功能相同, 但它们的区别在于, 用 KEEP 指令编程时, 要用 3 条语句, 当使用保持继电器 HR 作输出时, 具有掉电保持的功能; 用 SET 和 RSET 指令编程时, 要用 4 条语句, 但 SET 和 RSET 指令语句之间可插入别的指令, 使用比较灵活。当 SET 指令的操作数是保持继电器 HR 时具有掉电保持功能; 对于图 3.21, 编程时需要 4 条语句, 用 OUT 指令输出时无掉电保持功能。

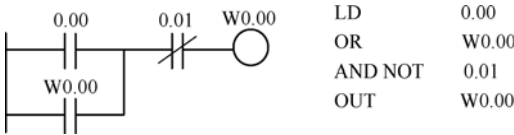


图 3.21 启、保、停控制程序

6. 上升沿微分 DIFU、下降沿微分 DIFD、条件上升沿微分 UP、条件下降沿微分 DOWN 指令

表 3.8 是微分分类指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。



表 3.8 微分类指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
DIFU(013) N 上升沿微分指令		范围：CIO、W、H、A 等 (除了 IR 中已作为输入通道的位) 以位为单位进行操作	当执行条件由 OFF 变为 ON 时，使指定的继电器 N ON 一个扫描周期 指令的执行结果不影响标志位
DIFD(014) N 下降沿微分指令			当执行条件由 ON 变为 OFF 时，使指定的继电器 N ON 一个扫描周期 指令的执行结果不影响标志位
UP(521) 条件上升沿微分指令		无	当执行条件由 OFF 变为 ON 时，输出 ON 一个扫描周期，连接到下一段 指令的执行结果不影响标志位
DOWN(522) 条件下降沿微分指令			当执行条件由 ON 变为 OFF 时，输出 ON 一个扫描周期，连接到下一段 指令的执行结果不影响标志位

使用 DIFU 和 DIFD 指令时要注意：

在第  $n$  次扫描时检测到输入条件为 OFF、第  $n+1$  次扫描检测到输入条件为 ON 时，DIFU 指令才会被执行。因此对于开机时就 ON 的执行条件，DIFU 指令不执行。同样对于开机时就 OFF 的执行条件，DIFD 指令也不执行。

图 3.22 使用了 DIFU 和 DIFD 指令，图中 TS 是扫描周期。在图 3.22 中，0.00 是 DIFU 和 DIFD 指令的执行条件。从 0.00 由 OFF 变为 ON 开始，100.00 只 ON 一个扫描周期；从 0.00 由 ON 变为 OFF 开始，100.01 只 ON 一个扫描周期。

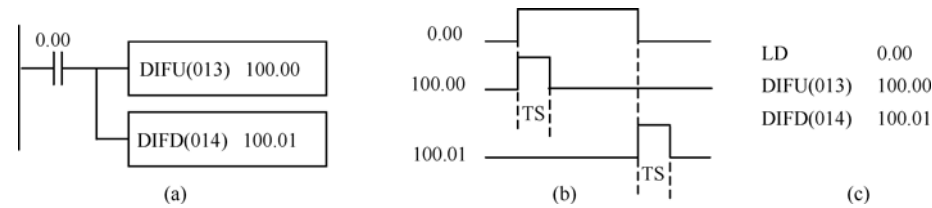


图 3.22 DIFU 和 DIFD 指令应用示例

DIFU 和 DIFD 指令常用在下面的几种场合：

- ① 利用 DIFU 和 DIFD 指令的操作位作为某指令的执行条件，使某条指令只在 DIFU 的执行条件由 OFF 变为 ON、或 DIFD 的执行条件由 ON 变为 OFF 时执行一次。
- ② 利用 DIFU 和 DIFD 指令产生脉冲信号。

上例中，当 0.00 闭合时，其上升沿使 100.00 闭合一个扫描周期，而后断开；当 0.00 断开时，其下降沿使 100.01 闭合一个扫描周期，而后断开；DIFU 和 DIFD 可分别由条件上升沿微分指令 UP(521) 和条件下降沿微分指令 DOWN(522) 替代，功能完全相同，见下面的例子。

UP 和 DOWN 指令应用示例如图 3.23 所示。

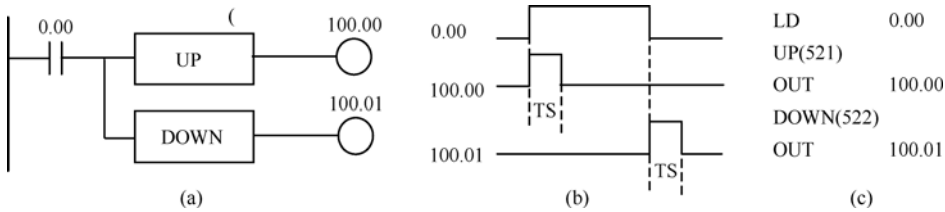


图 3.23 UP 和 DOWN 指令应用示例

3.3.2 指令的微分/非微分形式和即时刷新形式

1. 指令的微分/非微分形式

指令分为微分型和非微分型两种形式，CPlH 系列 PLC 的应用指令多数兼有这两种形式。微分型指令要在其助记符前加标记@（上升沿微分）或%（下降沿微分）。两种指令的区别是：对非微分型指令，只要其执行条件为 ON，则每个扫描周期都将执行该指令；上升沿微分型指令（指令前带“@”）仅在其执行条件由 OFF 变为 ON 时才执行一次，如果执行条件不发生变化，或者从上一个扫描周期的 ON 变为 OFF，该指令不执行；下降沿微分型指令（指令前带“%”）仅在其执行条件由 ON 变为时 OFF 才执行一次。如果执行条件不发生变化，或者从上一个扫描周期的 OFF 变为 ON，该指令都不执行。

CPlH 系列 PLC 只有少数几条指令可以附加下降沿微分型（指令前带%）功能，如 LD、AND、OR、SET、RSET。若其他指令需要仅在执行条件的下降沿执行时，要将执行条件与 DIFD 或 DOWN 指令组合使用。

2. 指令的即时刷新形式

指令的即时刷新形式是指在指令的助记符前加标记“!”，它的含义是 CPU 在当前的扫描周期，执行指令的前后，分别立即刷新 PLC 的输入和输出，而不是周期性刷新。

指令的微分形式和即时刷新形式两种功能可以组合使用。微分型与即时刷新型指令的指令名称、助记符、功能及刷新模式见表 3.9。

表 3.9 微分型与即时刷新型指令

指令名称	助记符	功能	刷新模式
常用指令	LD、LD NOT、AND、AND NOT、OR、OR NOT	CPU 在当前扫描周期读入指定触点的状态，下一个扫描周期产生结果	周期性刷新
	OUT、OUT NOT	指令执行后，将指定线圈的状态在本扫描周期 I/O 刷新阶段输出	
上升沿微分型指令	@LD、@AND、@OR 等	指定触点上升沿时执行指令，仅执行 1 个周期	
下降沿微分型指令	%LD、%AND、%OR、%SET、%RSET	指定触点下降沿时执行指令，仅执行 1 个周期	
即时刷新型指令	!LD、!LD NOT、!AND、!AND NOT、!OR、!OR NOT、!SET、!RSET	读入触点的 ON/OFF 状态后，立即执行指令	指令执行前刷新
	!OUT、!OUT NOT	执行指令后的结果立即输出给指定线圈	指令执行后刷新
上升沿微分即时刷新型指令	!@LD、!@AND、!@OR	读入指定触点的状态后，在其上升沿执行指令，仅执行 1 个周期	指令执行前刷新
下降沿微分即时刷新型指令	!%LD、!%AND、!%OR	读入指定触点的状态后，在其下降沿执行指令，仅执行 1 个周期	

注：1. 即时刷新型指令仅适应于 CPU 单元内置的输入/输出点，不能用于 CPlW 或 CPlMA 系列的扩展 I/O 单元；

2. 很多指令都有上升沿微分功能，但只有表中所列的指令具有下降沿微分型功能；

3. 除表中所列的即时刷新型指令，还有下列指令具有即时刷新功能：!KEEP、!DIFU、!DIFD、!CMP、!MOV、!CPS、!SETB、!RSTB、!OUTB。

3.4 时序控制指令

3.4.1 结束指令 END 和空指令 NOP

1. 结束指令 END (001)

程序的结尾处一定要安排 END 指令，因为 CPU 扫描到 END 指令时即认为程序到此结束，END 后面的程序一概不执行，并马上返回到程序的起始处再次扫描程序。若程序结束时没写 END 指令，

将出现程序错误。在调试程序时可以将 END 指令插在各段程序之后，对程序进行分段调试，调试结束时再删除插在中间各段程序之后的 END 指令。

图3.24中使用了 END 指令，请注意 END 指令的梯形图画法和语句的写法。图中，常闭触点 0.03 与上一行并联后再与常开触点 0.02 串联而形成一个触点组，0.04 与上面的触点组再并联。0.04 与上面的触点组两者中有一个为 ON，100.02 即为 ON。

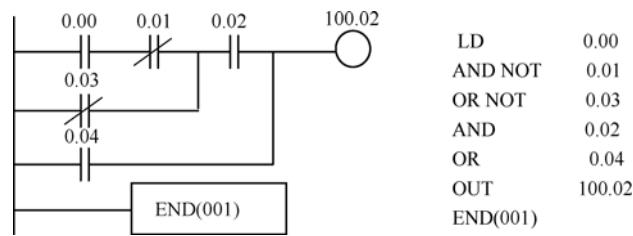


图 3.24 使用 END 指令的例子

本书在以后的梯形图程序中通常省略 END 指令，不再另行说明。

2. 空指令 NOP (000)

不具备任何功能(不进行程序处理)的指令。在梯形图中无表示。NOP 指令常用来修改程序。当修改程序删去一条指令时，后面语句的地址发生变化。而用 NOP 指令代替所删指令的位置，后面语句的地址保持不变。

例如，用 NOP 代替 AND N 语句，可把 AND 语句中的触点 N 短接，用 NOP 代替 OR N 语句，可把 OR 语句中的触点 N 断掉等，而其他语句的地址保持不变。

图3.25是使用 NOP 指令的例子。欲将图3.25(a)变成图3.25(b)的梯形图，将图3.25(a)语句表中的 AND 0.01 改写成 NOP (000) 即可。若欲去掉 LD 0.00，不仅把第一条语句处改写成 NOP (000)，还要将下一条语句 AND 0.01 改写成 LD 0.01，否则会出现语法错误。

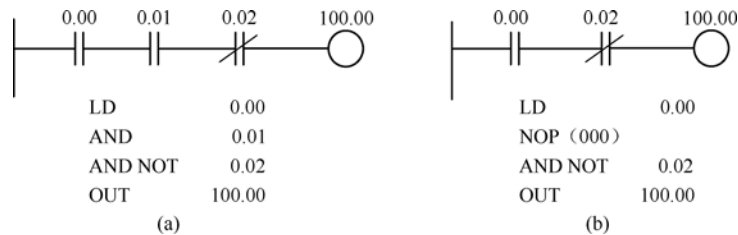


图 3.25 使用 NOP 指令的例子

3.4.2 互锁 IL/互锁解除 ILC 指令

IL/ILC 指令常用于控制程序的流向。表 3.10 是指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.10 IL/ILC 指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
IL (002) 互锁指令		无操作数	当 IL 的输入条件为 ON 时，IL 和 ILC 之间的程序正常执行； 当 IL 的输入条件为 OFF 时，IL 和 ILC 之间的程序不执行指令的执行结果不影响标志位
ILC (003) 互锁解除指令			

使用 IL/ILC 指令时应注意：

- ① 不论 IL 的输入条件是 ON 还是 OFF，CPU 都要对 IL/ILC 之间的程序段进行扫描。
- ② 如果 IL 的执行条件为 OFF，则位于 IL 和 ILC 之间的程序段不执行，此时 IL 和 ILC 之间各内部器件的状态如下。

所有 OUT 和 OUT NOT 指令的输出位为 OFF，所有定时器都复位，KEEP 指令的操作位、计数器、移位寄存器及 SET 和 RSET 指令的操作位都保持 IL 为 OFF 以前的状态。

③ IL 和 ILC 指令可以成对使用，也可以多个 IL 指令配一个 ILC 指令，但不准嵌套使用，如 IL—IL—ILC—ILC。

图3.26(a)中 A 为分支点，A 右侧有 2 个分支，且每个分支都有触点控制。这时要使用互锁指令编程。图3.26(a)也可以画成图3.26(b)的结构，两图的功能一样。

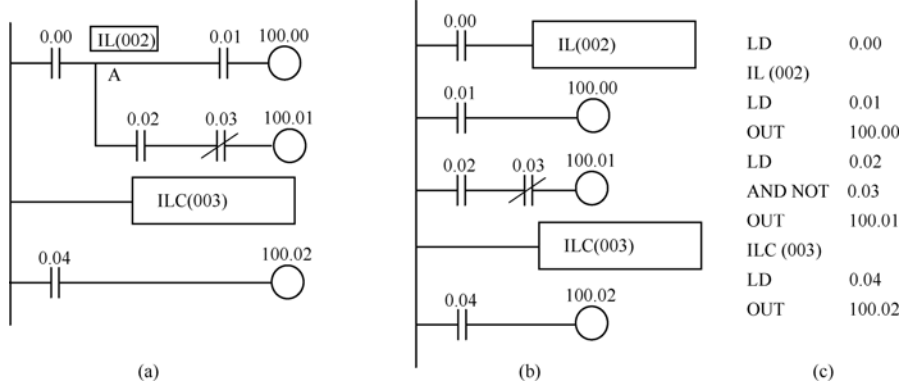


图 3.26 使用 IL/ILC 指令的例子

图3.26中，当 0.00 为 OFF 时，IL 和 ILC 之间的程序不执行，100.00、100.01 都处于 OFF 状态。当 0.00 为 ON 时，IL 和 ILC 之间的程序执行，100.00、100.01 的状态取决于各自分支上的控制触点的状态。

在图3.26中，若 A 右侧的第一分支中没有控制触点 0.01，就不必用互锁指令编程了。但如果把没有触点控制的分支放在下面，那也必须用互锁指令编程。

图3.27(a)的程序有两次分支，图3.27(a)也可以画成图3.27(b)的结构，两图的功能是一样的。图3.27(c)是它们的语句表。

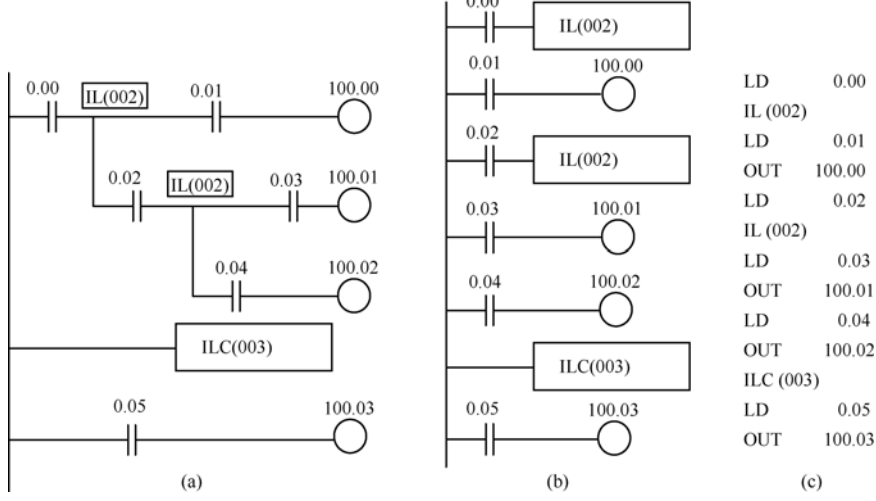


图 3.27 IL/ILC 指令使用方法举例

到此为止已经介绍了多种输出方式，归纳起来可以分为并联输出、连续输出和复合输出三种结构。三种输出方式的梯形图结构和语句表如图3.28所示，请注意各种输出方式语句表的写法。

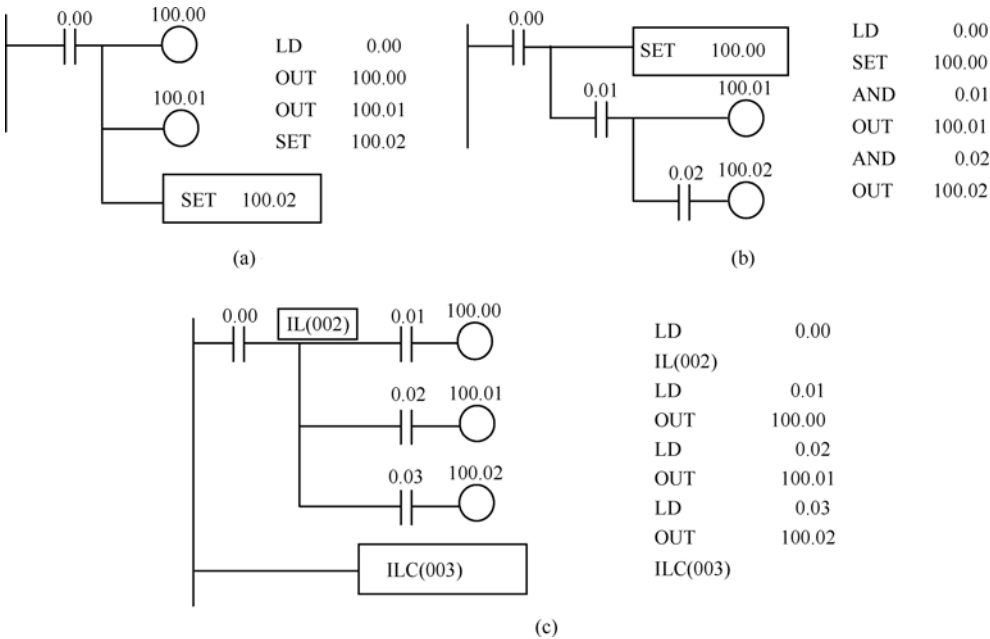


图 3.28 并联输出、连续输出和复合输出的程序结构

3.4.3 暂存继电器 TR

暂存继电器(TR)是继电器，而不是指令。暂存继电器可用来暂时存储当前指令执行的结果，所以处理梯形图的分支还有另外一种方法，即使用暂存继电器。

CPIH 系列 PLC 有编号为 TR0~TR15 的 16 个暂存继电器。如果某个 TR 位被设置在一个分支点处，则分支前面的执行结果就会存储在这个 TR 位中。对暂存继电器做如下说明：

- ① 在同一指令块中，同一 TR 位不能重复使用；
- ② TR 不是编程指令，而是继电器，只能配合 LD 或 OUT 指令一起使用。

图3.29为使用暂存继电器(TR)处理分支的例子。从语句表可以看出两种处理分支方法的区别：用 TR 时，是用 AND 指令连接下一个分支的触点；用 IL/ILC 时，是用 LD 指令连接下一个分支的触点。在分支多时，用 TR 处理分支程序比使用 IL/ILC 指令时语句表要烦琐一些。

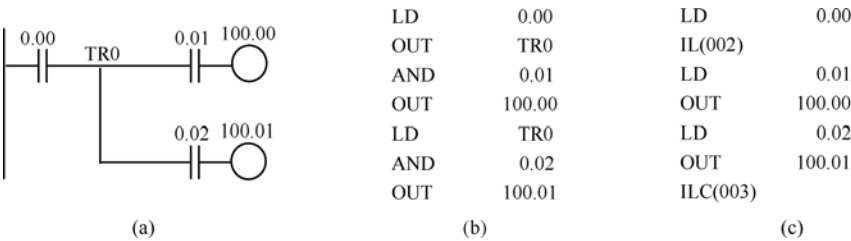


图 3.29 用 TR 与用 IL/ILC 处理分支程序的区别

图3.27程序有两次分支，用 TR 指令改写，如图3.30(a)所示，图3.30(b)为语句表程序。

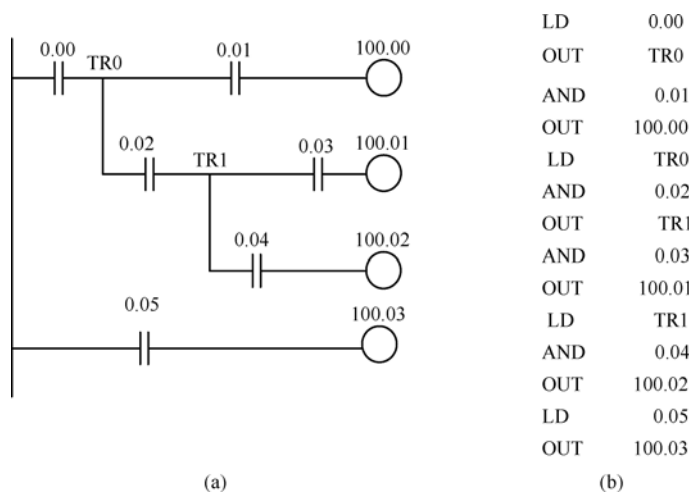


图 3.30 IL/ILC 指令使用方法举例

注意：采用计算机编程时，可直接用 CX-P 软件将梯形图转换为语句表，梯形图中的分支自动采用 TR 处理。

3.4.4 跳转 JMP/跳转结束 JME 指令

JMP/JME 是跳转和跳转结束指令，常用于控制程序的流向。表 3.11 是对这对指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.11 JMP 和 JME 指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
JMP(004) N 跳转指令		N 为跳转号，其范围为：#0000~00FF 或&0~255 JMP 的 N 可取的继电器区域：CIO、W、H、A、T/C、D、@D、*D、DR、IR 等 JME 的 N 只能取常数	当 JMP 的执行条件为 OFF 时，跳过 JMP 和 JME 之间的程序转去执行 JME 之后的程序；当 JMP 的执行条件为 ON 时，JMP 和 JME 之间的程序被执行 对标志位的影响： ① N 不在 0~255(0000~00FF)范围时 ER 为 ON ② JMP 没有同一编号的 JME 与之对应 ③ JMP 与对应同一编号的 JME 不在同一任务内
JME(005) N 跳转结束指令			

- 使用 JMP N 和 JME N 指令时注意：
- ① 发生转移时，JMP N 和 JME N 之间的程序不执行，且不占用扫描时间；
  - ② 发生跳转时，所有继电器、定时器、计数器均保持跳转前的状态不变，定时器定时继续；
  - ③ 不允许出现相同编号的两个以上的 JMP 指令或 JME 指令；
  - ④ 跳转指令可以嵌套使用，但必须是不同跳转号的嵌套，如 JMP &0—JMP &1—JME & 1—JME &0 等。

图 3.31 为使用 JMP/JME 的例子，0.00 是 JMP &0 的执行条件，当 0.00 为 OFF 时，JMP &0 到 JME &0 之间的程序不执行，而转去执行 JME &0 之后的程序，这时 100.00 和 101.00 保持转移前的状态。例如，若转移前 100.00 为 OFF，则转移期间也为 OFF，即使 0.01 为 ON；当 0.00 变为 ON 时，JMP &0 到 JME &0 之间的程序才被执行。

在两段程序切换时，常用到转移指令。如图 3.32 所示，当输入 0.00 为 ON 时，执行手动程序而不执行自动程序；当 0.00 为 OFF 时，跳过手动程序转去执行自动程序，请注意 JMP/JME 的这种用法。

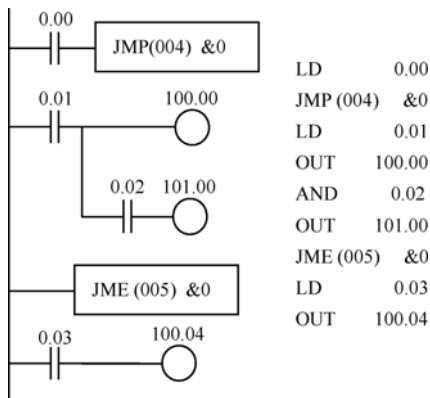


图 3.31 使用 JMP/JME 的例子

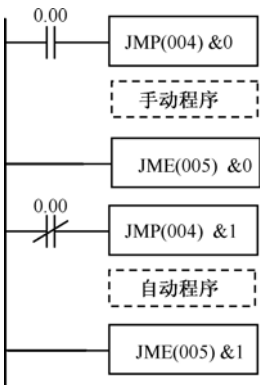


图 3.32 转移指令的用法

3.4.5 其他顺序控制指令

表 3.12 是其他顺序控制类指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.12 其他顺序控制类指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
CJP(510) N 条件转移指令		N: #0000~00FF 或&0~255 范围: CIO、W、H、A、T/C、D、@D、*D、DR 等	CJP 执行条件为 ON(CJPN 执行条件为 OFF)时,直接转移到同一编号的 JME 指令,转移时输出保持。CJP(CJPN)与 JME 成对使用 对标志位的影响: ① N 不在 0~255(0000~00FF)范围时 ER 为 ON ② CJP/ CJPN 没有同一编号的 JME 与之对应 ③ CJP/ CJPN 与对应同一编号的 JME 不在同一任务内
CJPN(511) N 条件非转移指令			
FOR(512) N 重复开始指令		N: 循环重复次数 #0000~00FF 或&0~255 范围: CIO、W、H、A、T/C、D、@D、*D、DR 等	无条件执行 FOR~NEXT 循环间的程序 N 次后,执行 NEXT 指令之后的程序; N 为&0 时,对 FOR~NEXT 间的程序不执行,而是执行 NOP 指令 对标志位的影响: 超过 FOR~NEXT 循环的最大嵌套层数(15)时, ER 为 ON
NEXT(513) 重复结束指令		无	
BREAK(514) 循环中断指令		无	当执行条件为 ON 时,强制结束正在运行的循环程序,在 FOR~NEXT 循环之间使用。1 条 BREAK 指令只能用于 1 个循环。可配置与 FOR~NEXT 循环数相同的 BREAK 指令、个数 对标志位的影响: ER、N、EQ 为 OFF
JMP0(515) 多重跳转指令		无	当执行条件为 ON 时,执行 JMP0~JME0 之间的程序;当执行条件为 OFF 时,跳过 JMP0~JME0 之间的程序,转去执行 JME0 之后的程序 在同一程序中可多次使用,但不可重叠、不可嵌套使用
JME0(516) 多重跳转结束指令			
MILH(517) N D 多重互锁指令(微分标志保持型)		D: 互锁编号 N: 互锁状态输出位 D 的范围: CIO、W、H、A、IR N 的范围: 0~15 以位为单位进行操作	当执行条件为 ON 时,各指令正常执行;当执行条件为 OFF 时,从该 MILH(或 MILR)到同一互锁号的 MILC 指令之间的输出互锁 MILH(或 MILR)与 MILC 可以嵌套使用 对标志位的影响: ER 为 OFF
MILR(518) N D 多重互锁指令(不保持型)			
MILC(519) N 多重互锁解除指令			

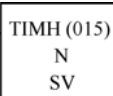
3.5 定时器/计数器指令

定时器 TIM 和计数器 CNT、CNTR 是经常使用的指令，下面举例说明它们的使用方法。

3.5.1 定时器 TIM/ TIMH 指令

表 3.13 是定时器指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.13 定时器指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
TIM N S 十进制定时器指令		N 是定时器编号，范围为：0~4095(十进制) SV 是定时器设定值(BCD #0000~9999) SV 范围为： CIO、W、H、A、T/C、D、@D、*D、#、DR 等	输入条件为 ON 时，定时开始，当前值从设定值开始每 0.1 s 减 1，直至为 0，定时时间到(定时时间为 S×0.1 s)，定时器的输出(定时完成标志)为 ON 且保持 当输入条件变为 OFF 时，定时器复位，定时器定时完成标志变为 OFF，并停止定时，其当前值 PV 恢复为 SV；定时器无掉电保持功能 对标志位的影响：当 SV 不是 BCD 数，ER 为 ON
TIMH(015) N S 十进制高速定时器指令			定时时间为 S×0.01 s，其余同 TIM 指令
TIMX(550) N S 十六进制定时器指令		设定值 SV 改为 BIN 方式：十六进制#0000~FFFF(十进制&0000~65535)，其余同上	设定值为十六进制，其余同 TIM 指令； 须做以下设置：在 CX-P 编程软件的工程工作区的项目“PLC”的下拉菜单的“属性”中，选以二进制的形式执行定时器/计数器

(1) 定时器的使用方法

在图 3.33 中，定时器 TIM0000 的设定值为#0050，当 0.00 为 OFF 时，TIM 0000 为复位状态，当前值 PV=#0050；自 0.00 为 ON 起 TIM0000 开始定时，其 PV 值从 0050 开始每隔 0.1 s 减去 1，减 50 次(5 s)时，PV 值减为 0000，此时定时完成标志 T0000 为 ON，其常开触点闭合，使 100.00 为 ON。若 0.00 一直为 ON，则定时完成标志 T0000 也一直保持 ON。若 0.00 变为 OFF，则 TIM0000 复位，PV 值恢复为设定值#0050，定时完成标志 T0000 为 OFF，100.00 变为 OFF。

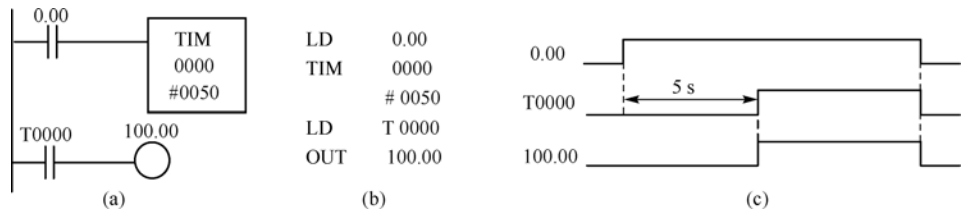


图 3.33 使用 TIM 指令的例子

(2) 定时器定时时间的扩展

一个定时器 TIM 的最大定时时间是 999.9 s，但几个定时器连用，则可获得更长的定时时间。如图 3.34 所示，以 TIM0000 的常开触点作为定时器 TIM0001 的执行条件，就可以实现定时器容量的扩展，总的定时时间为两个定时器 SV 值的和。当然也可以使用长时间定时器 TIML(542)实现长时间定时。TIML(542)的详细用法见 CP1H CPU 单元编程手册。



(3) 定时器的定时方式

虽然 TIM 是接通延时 ON 型的定时器，但经过合理编程，也可以实现接通延时 OFF、断开延时 ON、断开延时 OFF 的控制。如图 3.35 所示，从 0.00 为 ON 开始，100.00 经过 60 s 被接通(接通延时 ON)，而 100.01 则是经过 60 s 被断开(接通延时 OFF)。配合其他指令，读者可以练习用 TIM 指令编写出断开延时 ON 或断开延时 OFF 的定时控制的程序。

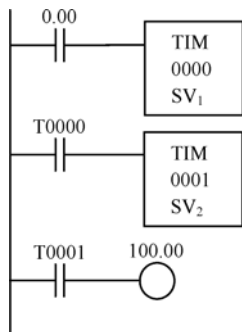


图 3.34 TIM 容量的扩展

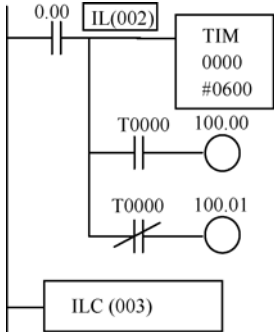


图 3.35 接通延时 ON 和接通延时 OFF 的控制

```
LD      0.00
IL (002)
TIM     0000
        #0600
LD      T0000
OUT     100.00
LD NOT  T0000
OUT     100.01
ILC (003)
```

使用定时器/计数器时应注意：

- ① 定时器的编号 0000～4095 由 TIM、TIMX(550)、TIMH(015)、TIMHX(551)、TIML(542)、TIMLX(553) 等指令共同占有，所以在同一程序中不同的指令最好不要使用同一个编号。
- ② 当 SV 为通道时，改变通道内的数据，其设定值即改变。也可以通过外部设备拨码器来改变其设定值。
- ③ 间接寻址 DM 通道不存在，是指以 DM 的内容为地址的通道不存在。
- ④ 定时器没有掉电保持功能，计数器有掉电保持功能。
- ⑤ 当扫描时间  $T_s > 0.1\text{ s}$  时，定时器 TIM 会不准确；当  $T_s > 0.01\text{ s}$  时，定时器 TIMH 会不准确。

3.5.2 计数器 CNT/ CNTR 指令

表 3.14 是计数器指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.14 计数器指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
<div>CNT N</div> <div>SV</div> <div>十进制</div> <div>计数器指令</div>		<div>N 是计数器编号，范围为： 0～4095(十进制)</div> <div>SV 是计数器设定值(BCD 计数器：#0000～9999)</div>	<div>CNT 功能：从 CP 端输入计数脉冲，当前值从设定值开始每来 1 个时钟脉冲减 1。当前值减为 0 时，计数满，其输出(计数完成标志)为 ON 且保持，并停止计数。只要复位端 R 为 ON，计数器即复位为 OFF 并停止计数，且当前值 PV 恢复为 SV</div> <div>计数器有掉电保持功能</div> <div>对标志位的影响：SV 不是 BCD 数时，ER 为 ON</div>
<div>CNTR(012) N</div> <div>SV</div> <div>十进制</div> <div>可逆循环计数器指令</div>		<div>N 是计数器编号，范围为： 0～4095(十进制)</div> <div>SV 是计数器设定值(BCD 计数器：#0000～9999)</div> <div>范围为： CIO、W、H、A、T/C、D、@D、*D、#、DR 等</div>	<div>只要复位 R 端为 ON，计数器即复位为 OFF 并停止计数，当前值 PV 为 0。从 ACP 和 SCP 端同时输入计数脉冲也不计数</div> <div>从 ACP 端输入计数脉冲为加计数；从 SCP 端输入计数脉冲为减计数；每当输入计数脉冲时，计数器加 1(或减 1)，加(或减)计数有进(或借)位时，输出(计数标志)ON 一个计数脉冲周期，之后再变为 OFF。而加(或减)计数器从 0(或 SV)再一次开始计数，循环下去</div> <div>可逆计数器有掉电保持功能</div> <div>对标志位的影响同上</div>

图3.36是 CNT 的工作波形，图3.37是 CNTR 加计数和减计数的工作波形。请注意观察各种计数器的计数方式、计数器的输出方式、复位时它们的当前值，并注意 CNTR 的循环计数过程。

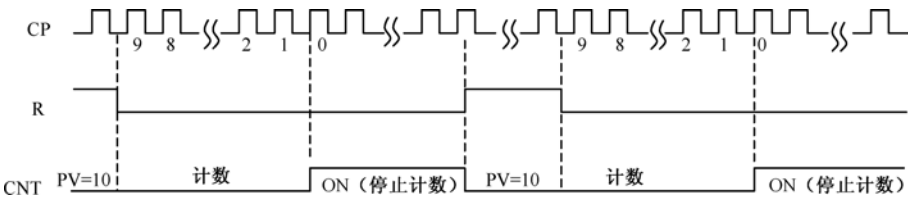


图 3.36 SV=10 CNT 的工作波形

由图3.37可见，用可逆计数器 CNTR 计数时，每个循环内计数的实际值比设定值多 1。

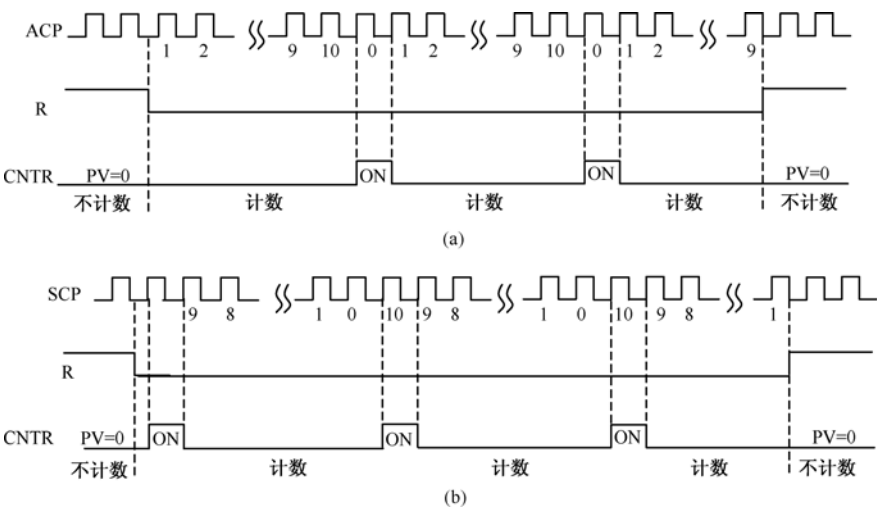


图 3.37 CNTR 的工作波形

1. 计数器 CNT

(1) 计数器 CNT 的计数功能

在图 3.38 中，CNT0000 的设定值为 200，表示设定值的数据是 200 通道的内容(设其中数据为 #0050)。当复位端 0.01 为 ON 时，计数器处于复位状态，计数完成标志 C0000 输出为 OFF。当复位端由 ON 变为 OFF 后计数器开始计数。其计数过程为：当 0.00 OFF→ON→OFF 一次(一个脉冲)，在脉冲的上升沿 CNT0000 的当前值 PV 就减 1。在当前值 PV 值减到 0000 时，也即计满 50 个脉冲时停止计数，此时计数完成标志 C0000 的变为 ON 且保持，C0000 的常开触点闭合，使 100.00 为 ON 且保持。若在计数过程中或在计满数以后，0.01 由 OFF 变为 ON，则计数器立即复位并停止计数。此时计数完成标志 C0000 变为 OFF，使 100.00 也变为 OFF。

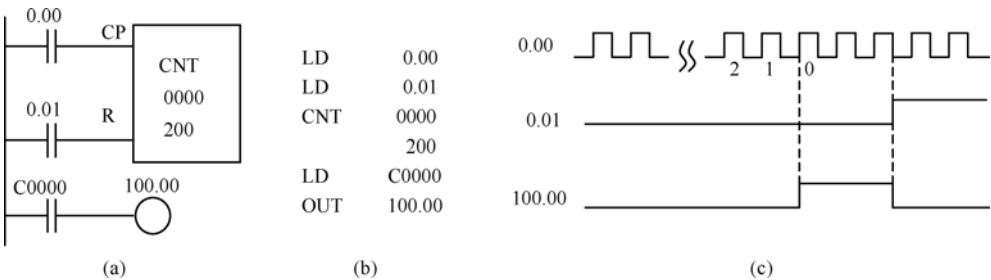


图 3.38 使用 CNT 指令的例子

### (2) 计数器 CNT 的定时功能

如果把图3.38中的 0.00 换成 P\_1s(产生秒脉冲, CF102), 则计数器又可以当定时器使用。例如, SV 为#0500, 当计数器计满 500 时, 其计数过程所用的时间刚好是 500 s。由于计数器有掉电保护功能, 所以用计数器做成的定时器也有掉电保持功能, 请注意 CNT 的这种用法。

### (3) 计数器 CNT 容量的扩展

用一个计数器的常开触点作为另一个计数器的计数输入, 即两个计数器连用, 就可以实现计数器容量的扩展, 总的计数器容量为两个计数器 SV 值的乘积, 如图 3.39 所示。图中, 用 P\_First\_Cycle(A200.11)对两个计数器进行初始复位, 计数过程中 CNT0000 能自复位。

## 2. 可逆计数器 CNTR

### (1) 可逆计数器 CNTR 的计数功能

如图3.40所示, 当复位端 0.03 为 ON 时 CNTR 0046 复位, 当前值变为 0000, 此时既不进行加计数、也不进行减计数。当 0.03 变为 OFF 时计数器开始计数, 其计数过程如下。

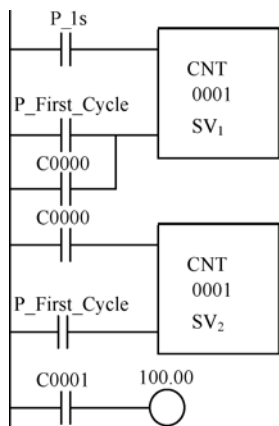


图 3.39 CNT 容量的扩展

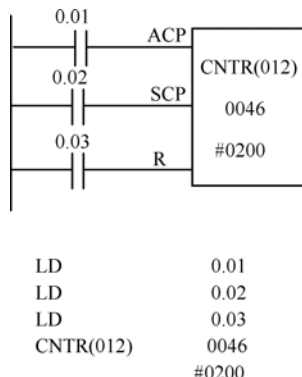


图 3.40 使用 CNTR 的例子

若 0.02 OFF、由 0.01 输入计数脉冲时为加计数器。0.01 每输入一个计数脉冲, CNTR 0046 的当前值加 1。当 PV=#0200 时, 再输入一个计数脉冲时, PV 值变为 0000(有进位), 计数完成标志 C0046 变为 ON。若再来一个计数脉冲时, PV=1, 计数完成标志 C0046 变为 OFF, 且开始下一个循环的计数。

若 0.01 OFF、由 0.02 输入计数脉冲时为减计数器。0.02 每输入一个计数脉冲, CNTR 0046 的当前值减 1, 当 PV=0000 时, 再输入一个计数脉冲时, PV 变为 0200(有借位), 同时计数完成标志 C0046 变为 ON。若再来一个计数脉冲时, PV=0199, 且计数完成标志 C0046 变为 OFF, 并开始下一个循环的计数。

当 0.01 和 0.02 同时输入计数脉冲时, 计数器不计数。

### (2) 可逆计数器的循环定时功能

图3.41中, SCP 端以 P\_Off(常 OFF)作为输入条件, 所以 CNTR0000 作为加计数器使用。ACP 端以 P\_1s 与 W0.00 的串联作为输入条件, 由 P\_1s 产生的秒脉冲作为计数脉冲输入, 此时计数器可作为定时器使用。R 端以 0.01 与 P\_First\_Cycle 的并联作为复位条件, 使 CNTR 0000 在 PLC 加电后的第一个扫描周期被复位。图中若 0.01 为 OFF, H0 中的数据是#0500, 请读者自行分析该图的功能。

### (3) 循环计数器容量的扩展

在图3.42中, CNTR0000 的计数完成标志常开触点连到 CNT0001 的计数脉冲输入端, 就可以构成大容量的循环计数器。例如, CNTR0000 指令的 H0 中若为#9999, CNT 0001 的 SV 为#1000, 则每经过 10000×1000 s, CNT 0001 的输出就会 ON 一次。请注意 CNT 和 CNTR 的编号方法。

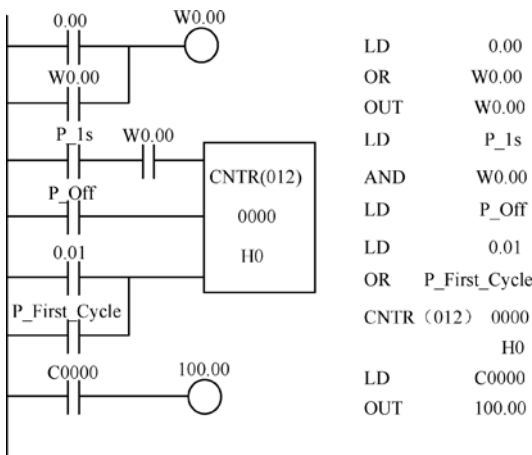


图 3.41 CNTR 的循环定时功能

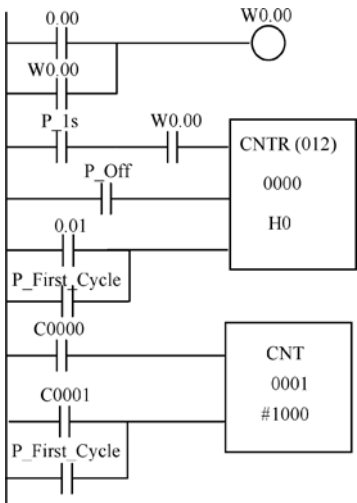


图 3.42 CNTR 容量的扩展

综上所述，CNT 和 CNTR 指令的主要区别是：

当计数器 CNT 达到设定值后，只要不复位，其输出(计数完成标志)就一直为 ON，即使计数脉冲仍在输入；计数器 CNTR 达到设定值后，其输出(计数完成标志)为 ON，只要不复位，在下一个计数脉冲到来时，计数器 CNTR 的输出(计数完成标志)立即变为 OFF，且开始下一轮计数，即 CNTR 是个循环计数器。

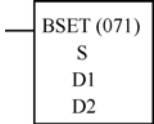
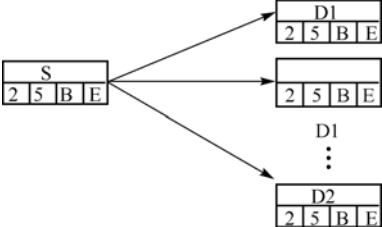
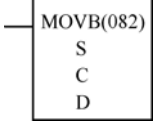
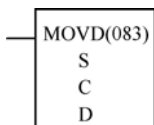
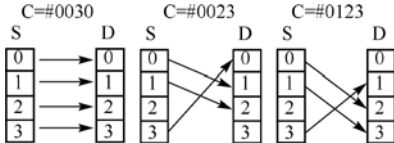
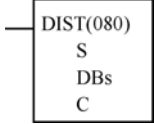

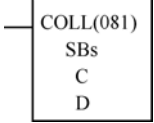
3.6 数据传送指令

CP1H 系列提供多种数据传送指令，表 3.15 是其指令的名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。表中所列数据传送类指令都有上升沿微分功能，为简洁只列出指令梯形图的原来形式。在指令梯形图的助记符前面加@就可成为指令梯形图的微分形式。

表 3.15 数据传送指令

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
传送指令 MOV/@ MOV (为即时刷新型!MOV)	<div><div>MOV(021)</div><div>S</div><div>D</div></div>	S 是源数据，范围：CIO、W、H、A、T/C、D、@D、*D、DR、#等 D 是目的通道，范围：CIO、W、H、A448~959、T/C、D、@D、*D、DR 等	当执行条件为 ON 时，将源数据 S 传送到通道 D 中对标志位的影响： ① 执行指令后若 D 中数据为 0000，EQ 为 ON ② 数据最高位为 1 时 N 为 ON
取反传送指令 MVN/@ MVN	<div><div>MVN(022)</div><div>S</div><div>D</div></div>		当执行条件为 ON 时，将源数据 S 按位取反后传送到通道 D 中 对标志位的影响同上
块传送指令 XFER/@ XFER	<div><div>XFER(070)</div><div>W</div><div>S</div><div>D</div></div>	W 是通道数，范围：CIO、W、H、A、T/C、D、@/*D、DR、#等 S 是源数据块开始通道，范围：CIO、W、H、A、T/C、D、@/*D、DR 等 D 是目的数据块开始通道，范围：CIO、W、H、A448~959、T/C、D、@/*D、DR 等 S+W-1 和 D+W-1 不能超出所在的区域	当执行条件为 ON 时，将连续 W 个通道中的数据对应传送到另外连续 W 个通道中去，如下图所示： <div><div><div>S</div><div>2 5 B E</div><div>S+1</div><div>0 8 B 9</div><div>S+2</div><div>9 3 2 5</div><div>⋮</div><div>S+W-1</div><div>7 3 C A</div></div><div><div>D</div><div>2 5 B E</div><div>D+1</div><div>0 8 B 9</div><div>D+2</div><div>9 3 2 5</div><div>⋮</div><div>D+W-1</div><div>7 3 C A</div></div></div> 对标志位的影响：指令执行时，将 ER 置 OFF

(续表)

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
块设置指令 BSET/@ BSET		<p>S 是源数据, 范围: CIO、W、H、A、T/C、D、@/*D、DR、#等</p> <p>D1 是目的数据开始通道, 范围: CIO、W、H、A448~959、T/C、D、@/*D、DR 等</p> <p>D2 是目的数据结束通道, 其范围同 D1</p> <p>D1 和 D2 必须在同一区域, 且 D1≤D2</p>	<p>当执行条件为 ON 时, 将源数据 S 传送到从 D1 到 D2 的所有通道中去, 如下图所示:</p>  <p>下列情况下, 标志位 ER 为 ON:</p> <p>① D1&gt;D2</p> <p>② D1 和 D2 不在同一区域</p>
位传送指令 MOVB/ @ MOVB		<p>S 是源数据, 范围: CIO、W、H、A、T/C、D、@D、*D、DR、#等</p> <p>C 是控制数据 (BCD), 范围: CIO、W、H、A、T/C、D、@D、*D、DR 等</p> <p>D 是目的通道, 范围: CIO、W、H、A448~959、T/C、D、@D、*D、DR 等</p>	<p>当执行条件为 ON 时, 根据 C 的内容, 将 S 中指定的某一位传送到 D 的指定位中</p> <p>C 的 bit00~bit07 指定 S 中的位号, bit08~bit15 指定 D 中的位号</p> <p>对标志位的影响: 当 C 指定的位不存在, ER 为 ON</p>
数字传送指令 MOVD/ @ MOVD		<p>S 是源数据, 范围: CIO、W、H、A、T/C、D、@D、*D、DR、#等</p> <p>C 是控制数据 (BCD), 范围比 S 缺少 #</p> <p>D 是目的通道, 范围比 S 缺少 #, 且 A 区为 A448~959</p> <p>C 的含义为:</p> <p>bit0~bit 3: S 中欲传送的第一个数字位的位号 (BCD: 0~3)</p> <p>bit4~bit7: S 中欲传送的数字位的位数 (0:1 位; 1:2 位; 2:3 位; 3:4 位)</p> <p>bit8~bit11: D 中接收第一个数字的位号</p> <p>bit12~bit 15 不用</p>	<p>当执行条件为 ON 时, 根据 C 的内容, 将 S 中指定的数字传送到 D 中指定的数字位中</p>  <p>对标志位的影响: 当 C 指定的位不存在, ER 为 ON</p>
单字数据分配指令 DIST/@ DIST		<p>S 是源数据, 范围: CIO、W、H、A、T/C、D、@D、*D、DR、#等</p> <p>DBs 是目的基准通道, 范围: CIO、W、H、A448~959、T/C、D、@D、*D 等</p> <p>C 是偏移数 (BCD), 范围同 S</p>	<p>将 S 的内容送到 (DBs+C) 确定的通道中</p> <p>对标志位的影响:</p> <p>① 当 S 的内容为 0000 时, EQ 为 ON</p> <p>② 当 S 的内容最高位为 1 时, N 为 ON</p> <p>③ ER 为 OFF</p>
数据交换指令 XCHG/@ XCHG		<p>E1 是交换数据 1, 范围: CIO、W、H、A448~959、T/C、D、@D、*D、DR 等</p> <p>E2 是交换数据 2, 其范围同 E1</p>	<p>当执行条件为 ON 时, 将 E1 与 E2 的内容进行交换</p> <p>指令的执行不影响标志位</p>
数据调用指令 COLL/@ COLL		<p>SBs 是源基准通道, 范围: CIO、W、H、A、T/C、D、@D、*D 等</p> <p>C 是控制数据 (BCD), 范围: CIO、W、H、A、T/C、D、@D、*D、DR、#等</p> <p>D 是目的通道, 范围: CIO、W、H、A448~959、T/C、D、@D、*D、DR 等</p>	<p>当执行条件为 ON 时, 将 SBs+C 的内容送到 D 通道中</p> <p>对标志位的影响:</p> <p>① 当传送数据为 0000 时, EQ 为 ON</p> <p>② 当传送数据最高位为 1 时, N 为 ON</p> <p>③ ER 为 OFF</p>

下面举例说明几个常用的传送指令的使用方法。

3.6.1 传送 MOV/@MOV、求反传送 MVN/@MVN 指令

图3.43中使用了 MOV 和@MVN 指令，0.00 是两个指令的执行条件。执行 MOV 指令时，把常数 #2000 传送到通道 H0 中去。@MVN 指令附加了上升沿微分功能，它只在执行条件 0.00 由 OFF 变为 ON 时执行一次，此后，即使 0.00 一直为 ON，@MVN 指令也不再执行。执行 @MVN 指令把十六进制数#2001 按位求反后再送到 H1 中。图3.44是执行 MOV 和 @MVN 指令的示意图。

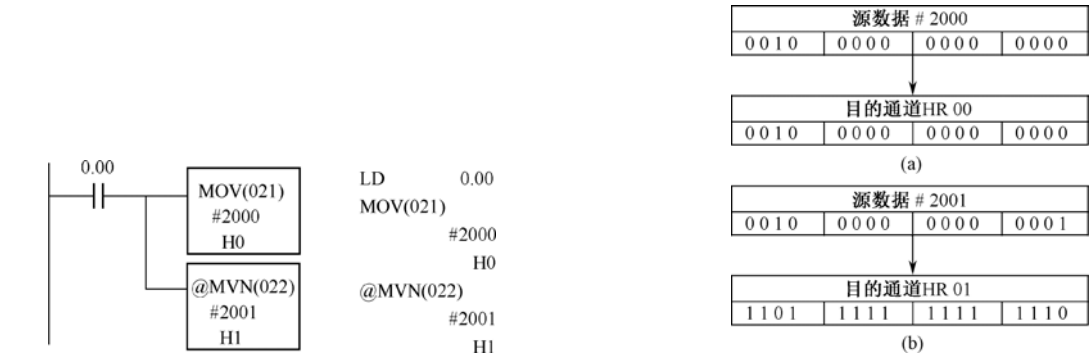


图 3.43 使用 MOV 和@MVN 指令的例子

图 3.44 执行 MOV 和@MVN 指令示意图

图3.45是使用 MOV 指令修改定时器设定值的程序。图中，当 0.00 为 ON、0.01 为 OFF 时，执行一次 MOV 指令将常数 0100 传送到 W0 中，所以 TIM 0000 的设定值为 10s，并开始定时。定时时间到，定时完成标志 T0000 变为 ON，100.00 也变为 ON。

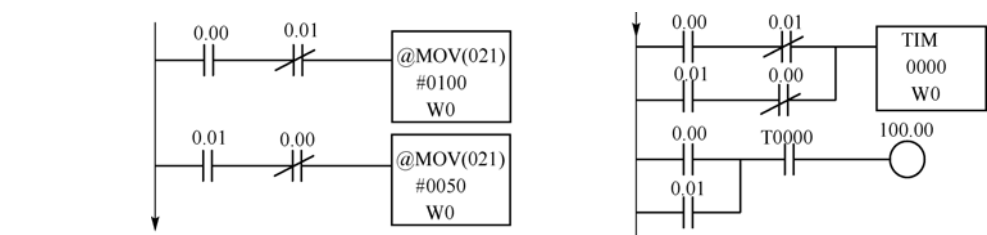


图 3.45 用 MOV 指令修改 TIM 的设定值

当需要改变定时器的定时值时，可令 0.00 为 OFF、0.01 为 ON，执行一次 MOV 指令将#0050 传送到 W0 中，于是 TIM 0000 的设定值就变为 5s。

由本例可见，当 TIM 用通道设置其设定值时，在程序执行过程中可以改变其设定值，请注意这种用法。

3.6.2 块设置指令 BSET

由表 3.15 中块设置指令的功能可知，执行一次 BSET/@BSET 指令，相当于执行了多次 MOV/@MOV 指令。因此当用 BSET/@BSET 指令向某个数据区传送 #0000 时，可将该区清零。

图3.46是使用 @BSET 指令的例子。图中 @BSET 指令的第二、第三操作数都是 T0，说明执行 @BSET 指令时，只把数据传送到 TIM0000 中。此处 @BSET 指令相当于 @MOV 指令的作用。该段程序的功能如下：

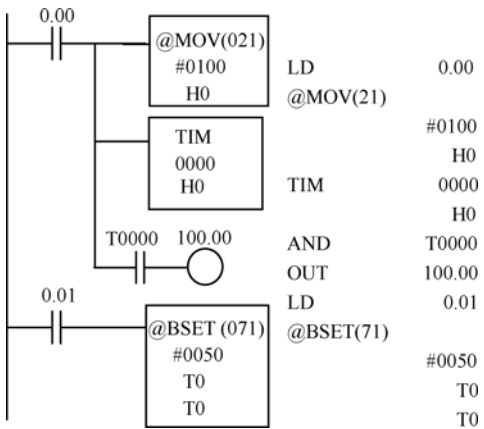


图 3.46 使用 @BSET 指令的例子

在 0.01 为 OFF、0.00 为 ON 时, 执行一次 MOV 指令将#0100 传送到通道 H0 中。自此定时器 TIM0000 就以设定值 100 (10 s) 开始定时, 每隔 0.1 s 当前值减 1, 经过 10 s, 定时完成标志 T0000 为 ON 且保持, 使 100.00 也变为 ON 且保持。

在程序执行过程中, 如果需要改变定时器 TIM0000 的当前值时, 可令 0.01 ON 一次, 通过执行 @BSET 指令来实现。例如, 在 TIM0000 的当前值为 0089 时 0.01 ON 一次, 执行一次 @BSET 指令将 #0050 传送到 TIM0000 中, TIM0000 的当前值立即变为 0050。自此, TIM0000 从 0050 开始每隔 0.1 s 当前值减 1, 直到当前值减为 0000 为止。由于 H0 中的数据没有改变, 在下一次定时器 TIM0000 工作时, 其定时值仍然是 0100。本例中, 执行 @BSET 指令只是改变了 TIM0000 的当前值。

用 @BSET 指令也可以改变 TIM 的设定值。本例中, 若 @BSET 指令的操作数不是 T0, 而是 H0, 则执行 @BSET 指令后, TIM0000 的设定值就变成 0050 了。



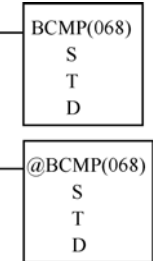
- 综上所述, MOV 和 BSET 指令的区别在于:
- ① 执行一次 MOV 指令, 只能向一个通道传送一个字; 而执行一次 BSET 指令, 可以向多个通道传送同一个字。
  - ② 用通道对 TIM/CNT 进行设定时, MOV 和 BSET 指令都可以改变 TIM / CNT 设定值。

3.7 数据比较指令

3.7.1 常用比较指令

表 3.16 是 CPIH 系列常用数据比较指令的名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.16 常用数据比较指令

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响																												
无符号比较指令 CMP (为即时刷新型!CMP)		S <sub>1</sub> 是比较数 1, S <sub>2</sub> 是比较数 2 S <sub>1</sub> 和 S <sub>2</sub> 范围: CIO、W、H、A、T/C、D、@D、*D、DR、#等	执行条件为 ON 时, 将比较 S <sub>1</sub> 和 S <sub>2</sub> , 并将比较结果送到各标志位, 见表: <table><tr><th>结果</th><th>GT</th><th>GE</th><th>EQ</th><th>LE</th><th>LT</th><th>NE</th></tr><tr><td>S<sub>1</sub>&gt;S<sub>2</sub></td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>S<sub>1</sub>=S<sub>2</sub></td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>S<sub>1</sub>&lt;S<sub>2</sub></td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	结果	GT	GE	EQ	LE	LT	NE	S <sub>1</sub> >S <sub>2</sub>	1	1	0	0	0	1	S <sub>1</sub> =S <sub>2</sub>	0	1	1	1	0	0	S <sub>1</sub> <S <sub>2</sub>	0	0	0	1	1	1
结果	GT	GE	EQ	LE	LT	NE																									
S <sub>1</sub> >S <sub>2</sub>	1	1	0	0	0	1																									
S <sub>1</sub> =S <sub>2</sub>	0	1	1	1	0	0																									
S <sub>1</sub> <S <sub>2</sub>	0	0	0	1	1	1																									
无符号双字比较指令 CMPL		S <sub>1</sub> 是比较数 1, S <sub>2</sub> 是比较数 2 S <sub>1</sub> 和 S <sub>2</sub> 范围: CIO、W、H、A、T/C、D、@D、*D、IR、#等	执行条件为 ON 时, 将 S <sub>1</sub> +1、S <sub>1</sub> 两个通道的内容与 S <sub>2</sub> +1、S <sub>2</sub> 两个通道的内容进行比较 对标志位的影响同上																												
无符号表格比较指令 BCMP/@ BCMP		S 是比较数据, 范围: CIO、W、H、A、T/C、D、@D、*D、DR、#等 T 是数据块的起始通道, 范围: 比 S 少 DR、# D 是比较结果通道, 范围: 比 S 少#	T~T+31 所指定的 32 字的表格, 分为 16 组的上限数据、下限数据 在执行条件为 ON 时, 将数据 S 与 16 组上、下限数据进行比较, 若 S 处在某组上、下限数据中, 则与该组对应的 D 通道的相应位为 ON, D 的对应位如下: <table><tr><td>T≤S≤T+1</td><td>D 的 bit 00</td></tr><tr><td>T+2≤S≤T+3</td><td>D 的 bit 01</td></tr><tr><td>T+4≤S≤T+5</td><td>D 的 bit 02</td></tr><tr><td>⋮</td><td>⋮</td></tr><tr><td>T+28≤S≤T+29</td><td>D 的 bit 14</td></tr><tr><td>T+30≤S≤T+31</td><td>D 的 bit 15</td></tr></table> 对标志位的影响: 当比较结果为 0 时, EQ 标志为 ON	T≤S≤T+1	D 的 bit 00	T+2≤S≤T+3	D 的 bit 01	T+4≤S≤T+5	D 的 bit 02	⋮	⋮	T+28≤S≤T+29	D 的 bit 14	T+30≤S≤T+31	D 的 bit 15																
T≤S≤T+1	D 的 bit 00																														
T+2≤S≤T+3	D 的 bit 01																														
T+4≤S≤T+5	D 的 bit 02																														
⋮	⋮																														
T+28≤S≤T+29	D 的 bit 14																														
T+30≤S≤T+31	D 的 bit 15																														

(续表)

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响																		
表格一致指令 TCMP/@ TCMP	<div><div>TCMP(085)</div><div>S</div><div>T</div><div>D</div></div> <div><div>@TCMP(085)</div><div>S</div><div>T</div><div>D</div></div>	<p>S 是比较数据，范围：CIO、W、H、A、T/C、D、@D、*D、DR、#等</p> <p>T 是数据块的起始通道，范围：比 S 少 DR、#</p> <p>D 是比较结果通道，范围：比 S 少#</p>	<p>T~T+15 的 16 通道数据构成表格</p> <p>在执行条件为 ON 时，将数据 S 与表中 16 通道的数据进行比较，若 S 与比较表中某个通道的数据相同，则与该通道对应的 D 通道的位为 ON，对应关系如下(设 S=0005)：</p> <table><tr><th>比较表</th><th>结果通道</th><th>对应位状态</th></tr><tr><td>H0 0101</td><td>H19.00</td><td>0</td></tr><tr><td>H1 0151</td><td>H19.01</td><td>0</td></tr><tr><td>H2 0005 →</td><td>H19.02</td><td>1</td></tr><tr><td>⋮</td><td>⋮</td><td>⋮</td></tr><tr><td>H15 0605</td><td>H19.15</td><td>0</td></tr></table> <p>对标志位的影响：当比较结果为 0 时，EQ 标志为 ON</p>	比较表	结果通道	对应位状态	H0 0101	H19.00	0	H1 0151	H19.01	0	H2 0005 →	H19.02	1	⋮	⋮	⋮	H15 0605	H19.15	0
比较表	结果通道	对应位状态																			
H0 0101	H19.00	0																			
H1 0151	H19.01	0																			
H2 0005 →	H19.02	1																			
⋮	⋮	⋮																			
H15 0605	H19.15	0																			

下面举例说明其中两种比较指令的使用方法。

1. 无符号比较指令 CMP

图3.47中使用了无符号比较指令 CMP。执行指令 CMP 时，将 TIM0000 的当前值与常数 0200 进行比较。程序的功能为：在 0.00 为 ON 时，TIM0000 开始定时，CMP 指令开始执行。由于每隔 0.1 s TIM0000 当前值减 1，所以在 0.00 为 ON 之后的一段时间内，若 TIM0000 的当前值大于 0200 时，P\_GT、100.00 为 ON；当 TIM0000 的当前值为 0200 时，P\_EQ、100.01 为 ON；当 TIM0000 的当前值小于 0200 时，P\_LT、100.02 为 ON；当 TIM0000 的当前值等于 0000 时，100.02 和 100.03 为 ON。由本例可见，配合指令 CMP，用一个定时器可以控制多个输出位。

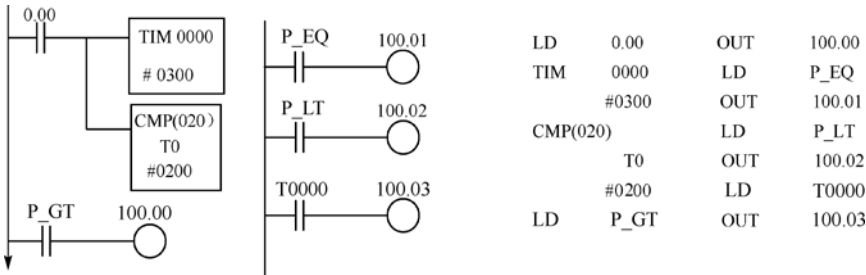


图 3.47 使用无符号比较指令 CMP 的例子

在执行条件为 ON 时，如果希望 CMP 指令只执行一次，可以使用 DIFU 或 DIFD 指令。如图3.48 所示，当 0.05 由 OFF 变为 ON 时，W0.00 ON 一个扫描周期，在此扫描周期里 CMP 指令执行。此后，即使 0.05 继续 ON，CMP 指令也不执行。

2. 无符号表格比较指令 BCMP/@BCMP

图3.49中使用了@BCMP 指令，图3.49(b)是执行@BCMP 指令后的结果。本例中表格由 D0~D31 组成。由于比较数据#1450 在 1401~1500 之间，所以与其对应的 H5.14 为 ON。

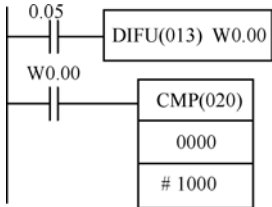


图 3.48 CMP 指令只执行一次



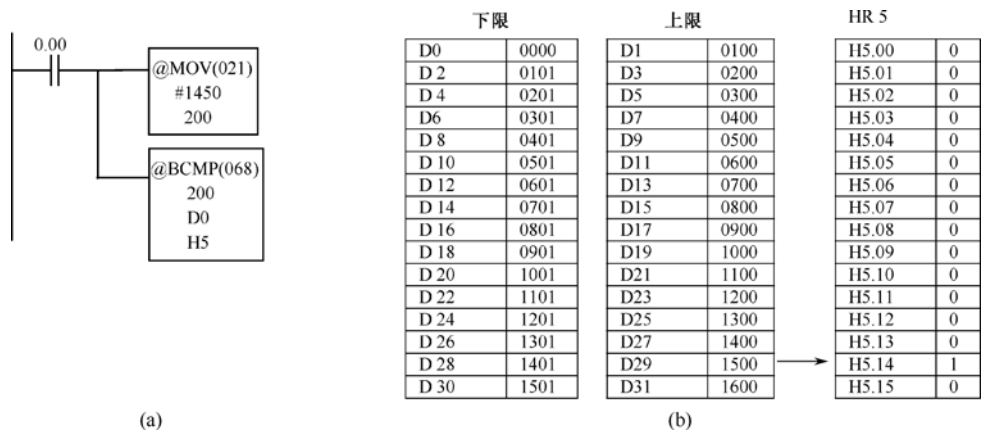


图 3.49 使用@BCMP 指令的例子

3.7.2 条件类比较指令

条件类比较指令是将指令的两个操作数的内容按条件进行比较，比较结果为真时，逻辑导通执行下一步程序。该类指令相当于一个接触器，满足条件时，接触器接通，不满足时断开，所以其逻辑连接方式分为 LD 型、AND 型和 OR 型三种。

表 3.17 是条件类比较指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.17 条件类比较指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响																																
条件 选项 S <sub>1</sub> S <sub>2</sub> 条件类比较指令	<div>条件选项 S<sub>1</sub> S<sub>2</sub></div>	条件是指比较条件，有：>、>=、=、<>、<=、< 选项是指无符号数(无)、带符号数(S)、字(无)或双字(L)比较 S <sub>1</sub> 是比较数 1，S <sub>2</sub> 是比较数 2 S <sub>1</sub> 和 S <sub>2</sub> 范围：CIO、W、H、A、T/C、D、@D、*D、DR(字)、IR(双字无符号)、#等	执行条件为 ON 时，将按条件比较 S <sub>1</sub> 和 S <sub>2</sub> ，并根据下表判断比较的结果，若比较结果为真，则此指令相当于一个继电器接通(为 ON) <table><tr><th>结果</th><th>字/双字(L)</th><th>字/双字(L)</th><th>字/双字(L)</th></tr><tr><td></td><td>比较数 1&gt;比较数 2</td><td>比较数 1=比较数 2</td><td>比较数 1&lt;比较数 2</td></tr><tr><td>&gt;</td><td>ON</td><td>OFF</td><td>OFF</td></tr><tr><td>&gt;=</td><td>ON</td><td>ON</td><td>OFF</td></tr><tr><td>=</td><td>OFF</td><td>ON</td><td>OFF</td></tr><tr><td>&lt;&gt;</td><td>ON</td><td>OFF</td><td>ON</td></tr><tr><td>&lt;=</td><td>OFF</td><td>ON</td><td>ON</td></tr><tr><td>&lt;</td><td>OFF</td><td>OFF</td><td>ON</td></tr></table>	结果	字/双字(L)	字/双字(L)	字/双字(L)		比较数 1>比较数 2	比较数 1=比较数 2	比较数 1<比较数 2	>	ON	OFF	OFF	>=	ON	ON	OFF	=	OFF	ON	OFF	<>	ON	OFF	ON	<=	OFF	ON	ON	<	OFF	OFF	ON
结果	字/双字(L)	字/双字(L)	字/双字(L)																																
	比较数 1>比较数 2	比较数 1=比较数 2	比较数 1<比较数 2																																
>	ON	OFF	OFF																																
>=	ON	ON	OFF																																
=	OFF	ON	OFF																																
<>	ON	OFF	ON																																
<=	OFF	ON	ON																																
<	OFF	OFF	ON																																

条件类比较指令选不同的条件和选项，可以组合成一大类指令，指令码为 300~328。例如，条件分别选>、>=、=、<>、<=、<，选项选无符号数，得到指令>(320)、>=(325)、=(300)、<>(305)、<=(316)、<(310)；若选项选带符号数，得到指令>S(320)、>=S(327)、=S(302)、<>S(307)、<=S(317)、<S(312)。

图3.50中使用了条件比较类指令。首先让执行条件 0.00 为 ON，用传送指令将 H10 赋值#8012，H20 赋值#2B1A。再让执行条件 0.01 为 ON，执行无符号条件比较“<”时，H10(&32786)与 H20(&11034)比较，此条件不满足，100.00 为 OFF；执行有符号条件比较“<S”时(最高位为符号位)，此时 H10(&-32750)与 H20(&+11034)比较，此条件满足，100.01 为 ON。

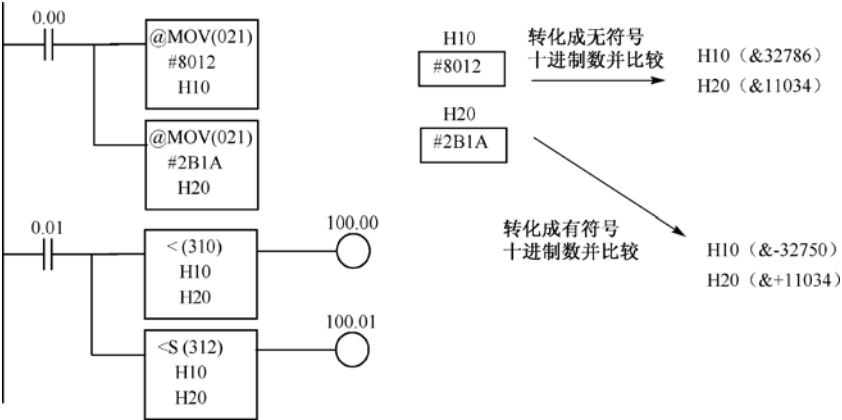


图 3.50 条件类比较指令示例

3.7.3 时刻类比较指令

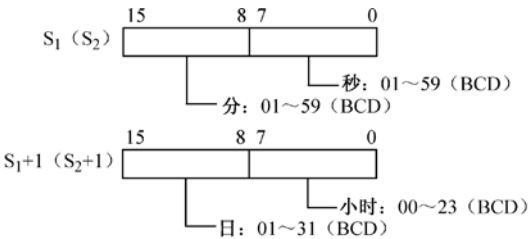
时刻类比较指令是将指令的两个时刻操作数的内容按条件进行比较，比较结果为真时，逻辑导通执行下一步程序。该类指令相当于一个接触器，满足条件时，接触器接通，不满足时断开，所以其逻辑连接方式分为 LD 型、AND 型和 OR 型三种。

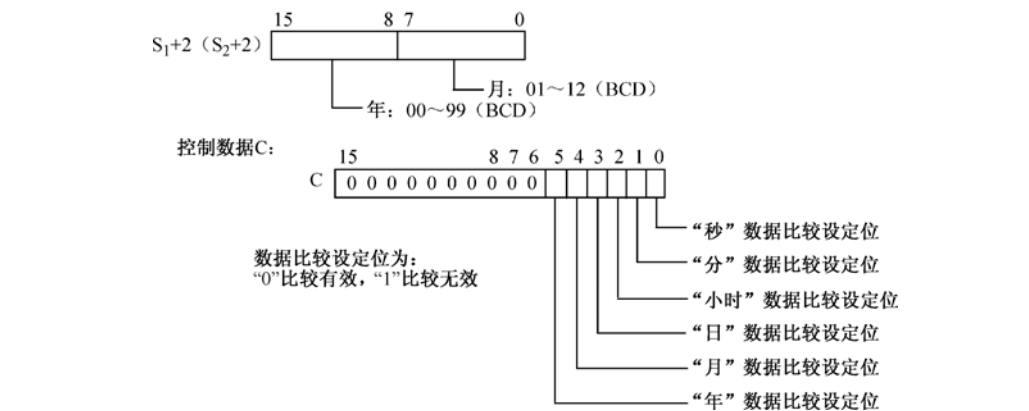
表 3.18 是时刻类比较指令的格式/名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。

表 3.18 时刻比较类指令

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响																												
条件 C S <sub>1</sub> S <sub>2</sub> 时刻类比较指令	<div>条件 C S<sub>1</sub> S<sub>2</sub></div>	条件是指比较条件，有：>DT、>=DT、=DT、<>DT、<=DT、<DT C 是控制数据，范围：CIO、W、H、A、T/C、D S <sub>1</sub> 是时刻 1 首通道 S <sub>2</sub> 是时刻 2 首通道 S <sub>1</sub> 和 S <sub>2</sub> 范围：CIO、W、H、A、T/C、D、@/*D	执行条件为 ON 时，将按条件比较 S <sub>1</sub> ~S <sub>1</sub> +2 的内容(时刻 1)和 S <sub>2</sub> ~S <sub>2</sub> +2 的内容(时刻 2)，并根据下表判断比较的结果，若比较结果为真，则此指令相当于一个继电器接通(为 ON) <table><tr><th>结果</th><th>时刻 1&gt; 时刻 2</th><th>时刻 1= 时刻 2</th><th>时刻 1&lt; 时刻 2</th></tr><tr><td>&gt;DT</td><td>ON</td><td>OFF</td><td>OFF</td></tr><tr><td>&gt;=DT</td><td>ON</td><td>ON</td><td>OFF</td></tr><tr><td>=DT</td><td>OFF</td><td>ON</td><td>OFF</td></tr><tr><td>&lt;&gt;DT</td><td>ON</td><td>OFF</td><td>ON</td></tr><tr><td>&lt;=DT</td><td>OFF</td><td>ON</td><td>ON</td></tr><tr><td>&lt;DT</td><td>OFF</td><td>OFF</td><td>ON</td></tr></table>	结果	时刻 1> 时刻 2	时刻 1= 时刻 2	时刻 1< 时刻 2	>DT	ON	OFF	OFF	>=DT	ON	ON	OFF	=DT	OFF	ON	OFF	<>DT	ON	OFF	ON	<=DT	OFF	ON	ON	<DT	OFF	OFF	ON
结果	时刻 1> 时刻 2	时刻 1= 时刻 2	时刻 1< 时刻 2																												
>DT	ON	OFF	OFF																												
>=DT	ON	ON	OFF																												
=DT	OFF	ON	OFF																												
<>DT	ON	OFF	ON																												
<=DT	OFF	ON	ON																												
<DT	OFF	OFF	ON																												

S<sub>1</sub>~S<sub>1</sub>+2 和 S<sub>2</sub>~S<sub>2</sub>+2 的含义：





CPU 单元的内部时钟数据通道为 A351~A353, 可以作为当前时刻的赋值通道。图3.51 为时刻比较类指令示例。由于控制通道 H10 的值是 0038 H, 表明仅对“秒”、“分”和“小时”的设定时刻和当前时刻进行比较。执行条件 0.00 为 ON 时, 将 H100 和 H101 通道的值(图中阴影范围)与 A351 和 A353 的 CPU 内部时钟当前时刻值进行等值比较, 当内部时钟到达设定时刻 10:00:00 时, “=DT”指令后的逻辑行导通, 100.00 为 ON。

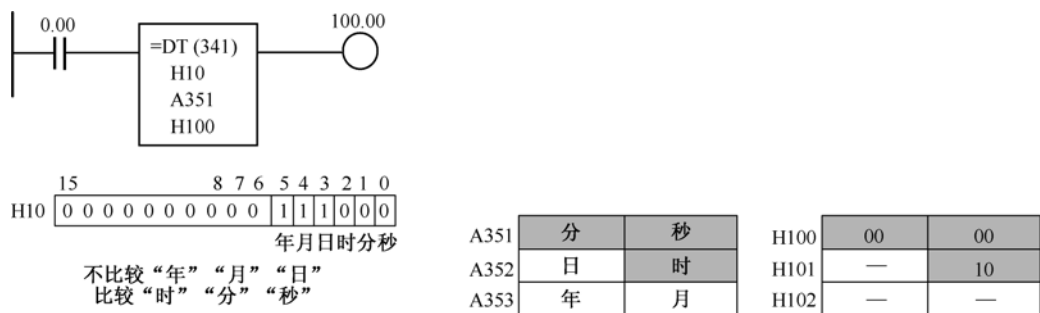


图 3.51 时刻比较类指令示例

3.7.4 其他比较指令

表 3.19 为其他数据比较类指令的名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响, 具体用法见欧姆龙 CP1H 编程手册。

表 3.19 其他数据比较指令

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响																												
带符号比较指令 CPS (有即时刷新功能: ! CPS)	<div><div>CPS(114)</div><div>S<sub>1</sub> S<sub>2</sub></div></div>	S <sub>1</sub> 和 S <sub>2</sub> 是带符号的比较数 1 和 2, 数字最高位为符号位 S <sub>1</sub> 和 S <sub>2</sub> 范围: CIO、W、H、A、T/C、D、@D、*D、DR、#等	执行条件为 ON 时, 比较 S <sub>1</sub> 和 S <sub>2</sub> 的内容 对标志位的影响: <table><tr><th>结果</th><th>GT</th><th>GE</th><th>EQ</th><th>LE</th><th>LT</th><th>NE</th></tr><tr><td>S<sub>1</sub>&gt;S<sub>2</sub></td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>S<sub>1</sub>=S<sub>2</sub></td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>S<sub>1</sub>&lt;S<sub>2</sub></td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	结果	GT	GE	EQ	LE	LT	NE	S <sub>1</sub> >S <sub>2</sub>	1	1	0	0	0	1	S <sub>1</sub> =S <sub>2</sub>	0	1	1	1	0	0	S <sub>1</sub> <S <sub>2</sub>	0	0	0	1	1	1
结果	GT	GE	EQ	LE	LT	NE																									
S <sub>1</sub> >S <sub>2</sub>	1	1	0	0	0	1																									
S <sub>1</sub> =S <sub>2</sub>	0	1	1	1	0	0																									
S <sub>1</sub> <S <sub>2</sub>	0	0	0	1	1	1																									
带符号双字比较指令 CPSL	<div><div>CPSL(115)</div><div>S<sub>1</sub> S<sub>2</sub></div></div>	S <sub>1</sub> 是比较数 1, S <sub>2</sub> 是比较数 2 S <sub>1</sub> 和 S <sub>2</sub> 范围: CIO、W、H、A、T/C、D、@D、*D、#等	执行条件为 ON 时, 将 S <sub>1</sub> +1、S <sub>1</sub> 两个通道的内容与 S <sub>2</sub> +1、S <sub>2</sub> 两个通道的内容进行比较 对标志位的影响同上																												

(续表)

格式/名称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
多字比较指令 MCMP	<div>MCMP(019) S<sub>1</sub> S<sub>2</sub> D</div>	S <sub>1</sub> 是比较数 1 首通道 S <sub>2</sub> 是比较数 2 首通道 D 比较结果输出通道 S <sub>1</sub> 和 S <sub>2</sub> 范围: CIO、W、H、A、T/C、 D、@D、*D 等 D 范围比比比较数范围多 DR	执行条件为 ON 时, 将分别以 S <sub>1</sub> 和 S <sub>2</sub> 为首通道的 16 个通道的内容进行比较, 比较结果反映到结果输出通道的对应位中 对标志位的影响: 当比较结果为 0 时, EQ 标志为 ON
扩展表比较指令 BCMP2	<div>BCMP2(502) S T D</div>	S 是比较数据 T 是比较表首通道 D 比较结果输出通道 S 范围: CIO、W、H、A、T/C、D、 @D、*D、DR 等 T、D 范围: CIO、W、H、A、T/C、 D、@D、*D、等	执行条件为 ON 时, 将比较数据 S 分别与 T 通道设定的若干组限值区间进行逐一比较, 比较结果反映到结果输出通道的对应位上 对标志位的影响: ER 标志为 OFF
区域比较指令 ZCP	<div>ZCP(088) S T1 T2</div>	S 是比较数据 T1 下限值首通道 T2 上限值首通道 S、T1、T2 范围: CIO、W、H、A、 T/C、D、@D、*D、DR、#等	执行条件为 ON 时, 将比较数据 S 分别与设定的上下限值进行比较, 比较结果反映到状态标志位中 状态标志位的变化同 CPS 指令
双字区域比较指令 ZCPL	<div>ZCPL(116) S T1 T2</div>	S 是比较数据(双字) T1 下限值首通道 T2 上限值首通道 S、T1、T2 范围: CIO、W、H、A、 T/C、D、@D、*D、IR、#等	执行条件为 ON 时, 将双字比较数据 S 分别与设定的上下限值进行双字比较, 比较结果反映到状态标志位中 状态标志位的变化同 CPS 指令

注: 1. 带符号比较是将比较数 1 和 2 作为带符号数看, 最高位为符号位;  
2. 双字(L)比较是将 S<sub>1</sub>+1、S<sub>1</sub> 两个通道的内容与 S<sub>2</sub>+1、S<sub>2</sub> 两个通道的内容比较;

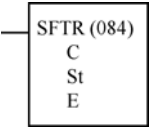
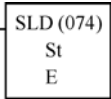
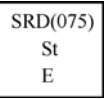
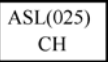
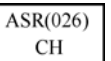
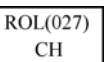
3.8 数据移位指令

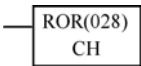
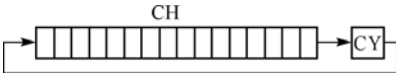
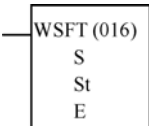
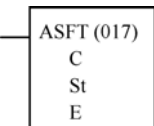
表 3.20 是数据移位指令的名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。下面说明几种常用的数据移位指令的使用方法。表中所列数据移位类指令除 SFT(010)外, 其他都有上升沿微分功能。为简洁只列出梯形图指令的原来形式。

表 3.20 数据移位指令(除 SFT 指令外, 其他指令都有上升沿微分功能)

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
移位寄存器指令 SFT	<div><div>IN SP R</div><div>SFT(010) St E</div></div> <div>IN 是数据输入端, SP 是移位脉冲输入 端, R 是复位端</div>	St 是移位开始通道号 E 是移位结束通道号 St、E 范围: CIO、W、H、A 等	当复位端 R 为 OFF 时, 在 SP 端的每个移位脉冲的上升沿时刻, St 到 E 通道中的所有数据按位依次左移一位。E 通道中数据的最高位溢出丢失, St 通道中的最低位则移进 IN 端的数据; 当复位端 R 为 ON 时, St 到 E 所有通道均复位为零, 且移位指令不执行 对标志位的影响: 间接寻址 IR 指定时, 位于 CIO、W、H、A 区之外, ER 为 ON

(续表)

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
左右移位寄存器指令 SFTR/@ SFTR		<p>C 为控制通道号, 范围: CIO、W、H、T/C、A、D、@/*D、DR 等</p> <p>C 通道中 bit12~bit15 的含义为:</p> <div data-bbox="463 363 738 592"></div> <p>St 是移位开始通道号 E 是移位结束通道号 St、E 范围: CIO、W、A448~959、H、T/C、D、@/*D 等</p>	<p>在执行条件为 ON 时, SFTR/@SFTR 指令执行。其功能为:</p> <p>① 控制通道 C 之 bit15 为 1 时, St 到 E 通道中所有数据及进位位 CY 全部清为 0, 且不接收输入数据</p> <p>② 控制通道 C 之 bit15 为 0 时, 在移位信号 bit14 ON 的作用下, 根据 C 之 bit12 的状态进行左移或右移</p> <p>左移: 从 St 到 E 通道的所有数据, 每个扫描周期按位依次左移一位, C 中 bit13 的数据移入开始通道 St 的最低位中, 结束通道 E 最高位的数据移入进位位 CY 中</p> <p>右移: 从 E 到 St 通道所有数据, 每个扫描周期按位依次右移一位, C 中 bit13 数据移入 E 的最高位中, St 最低位数据移入进位位 CY 中</p> <p>对标志位的影响:</p> <p>① St&gt;E 时, ER 为 ON</p> <p>② 移位溢出位进入 CY 中</p> <p>③ 复位输入为 1 时 CY 为 OFF</p>
数字左移指令 SLD@/ SLD		<p>St 是移位开始通道号 E 是移位结束通道号 St、E 范围: CIO、W、A448~959、H、T/C、D、@/*D 等</p>	<p>执行条件为 ON 时, E 到 St 通道的数据以 1 数字 (4bit) 为单位左移一次。0 进入 St 的最低数字位, E 中的最高位数字溢出丢失</p> <div data-bbox="807 833 1139 883"></div> <p>对标志位的影响: St&gt;E 时, ER 为 ON</p>
数字右移指令 SRD/@SRD		<p>St 是移位开始通道号 E 是移位结束通道号 St、E 范围: CIO、W、A448~959、H、T/C、D、@/*D 等</p>	<p>在执行条件为 ON 时, E 到 St 通道的数据以 1 数字 (4bit) 为单位右移一次。0 进入 E 的最高位, St 中的最低位数字溢出丢失</p> <div data-bbox="807 1024 1139 1073"></div> <p>对标志位的影响: St&gt;E 时, ER 为 ON</p>
位左移指令 ASL/@ASL		<p>CH 是移位通道号 范围: CIO、W、A448~959、H、T/C、D、@D、*D、DR 等</p>	<p>执行条件为 ON 时, 每执行一次移位指令, 将 CH 通道中的数据按位左移一位, 最高位移到 CY 中, 0 移进最低位</p> <p>对标志位的影响:</p> <p>① 移位溢出的位进入 CY</p> <p>② CH 中的内容为 0000 时, EQ 为 ON</p> <p>③ 移位结果最高位为 1 时, N 为 ON</p>
位右移指令 ASR/@ASR		<p>CH 是移位通道号 范围: CIO、W、A448~959、H、T/C、D、@D、*D、DR 等</p>	<p>执行条件为 ON 时, 每执行一次移位指令, 将 CH 通道中的数据按位右移一位, 最低位移到 CY 中, 0 移进最高位</p> <p>对标志位的影响:</p> <p>① 移位溢出的位进入 CY</p> <p>② CH 中的内容为 0000 时, EQ 为 ON</p> <p>③ 负标志 N 为 OFF</p>
带 CY 循环左移位指令 ROL/@ROL		<p>CH 是移位通道号 范围: CIO、W、A448~959、H、T/C、D、@D、*D、DR 等</p>	<p>执行条件为 ON 时, 每执行一次移位指令, 将 CH 通道中的数据连同 CY 的内容, 按位循环左移一位, 其过程如下图所示</p> <div data-bbox="778 1652 1181 1728"></div> <p>对标志位的影响同 ASL 指令</p>

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响																																																																
带 CY 循环右移位指令 ROR/@ROR		CH 是移位通道号 范围：CIO、W、A448~959、H、T/C、D、@D、*D、DR 等	执行条件为 ON 时，每执行一次移位指令，将 CH 通道中的数据连同 CY 的内容，按位循环右移一位，其过程如下图所示  对标志位的影响同 ASL 指令																																																																
字移位指令 WSFT/@WSFT		S 是数据 St 是移位开始通道号 E 是移位结束通道号 S 范围：CIO、W、H、A、T/C、D、@/*D、DR、#等 St、E 范围：CIO、W、A448~959、H、T/C、D、@/*D 等	执行条件为 ON 时，每执行一次移位指令，S、E 和 St 通道中的数据以字为单位左移一位，E 通道数据溢出丢失，S 移入到 St 通道中 对标志位的影响：St>E，ER 为 ON																																																																
非同步移位寄存器指令 ASFT/@ASFT		C 是控制数据，范围：CIO、W、H、A、T/C、D、@/*D、DR 等 C 的含义为： bit13，移位方向 (1，下移；0，上移) bit14，是否允许移位 (1，允许；0，不允许) bit15，是否复位 (1，复位；0，正常操作) 其余位为 0 St 是移位开始通道号 E 是移位结束通道号 St、E 范围：CIO、W、A448~959、H、T/C、D、@/*D 等	由 St 和 E 之间通道组成移位寄存器。当执行条件为 ON 时，根据 C 的内容，将所有数据为 0000 的通道与相邻通道进行数据交换，最后将所有 0000 集中在寄存器的上半部分或下半部分 上移时，所有数据为 0000 的通道与紧邻的高地址通道进行数据交换；下移时，所有数据为 0000 的通道与紧邻的低地址通道进行数据交换 例如，C=#6000 St=D0 E=D7，移位过程如下图所示： <div><div><div>执行前</div><table><tr><td>D0</td><td>0000</td></tr><tr><td>D1</td><td>0000</td></tr><tr><td>D2</td><td>507A</td></tr><tr><td>D3</td><td>2B68</td></tr><tr><td>D4</td><td>0000</td></tr><tr><td>D5</td><td>0234</td></tr><tr><td>D6</td><td>5147</td></tr><tr><td>D7</td><td>0000</td></tr></table></div><div><div>执行后</div><table><tr><td>D0</td><td>0000</td></tr><tr><td>D1</td><td>507A</td></tr><tr><td>D2</td><td>0000</td></tr><tr><td>D3</td><td>2B68</td></tr><tr><td>D4</td><td>0234</td></tr><tr><td>D5</td><td>0000</td></tr><tr><td>D6</td><td>5147</td></tr><tr><td>D7</td><td>0000</td></tr></table></div></div> <div>第一次执行指令</div> <div><div><div>执行前</div><table><tr><td>D0</td><td>0000</td></tr><tr><td>D1</td><td>5070</td></tr><tr><td>D2</td><td>0000</td></tr><tr><td>D3</td><td>2468</td></tr><tr><td>D4</td><td>0234</td></tr><tr><td>D5</td><td>0000</td></tr><tr><td>D6</td><td>5147</td></tr><tr><td>D7</td><td>0000</td></tr></table></div><div><div>执行后</div><table><tr><td>D0</td><td>5070</td></tr><tr><td>D1</td><td>0000</td></tr><tr><td>D2</td><td>2468</td></tr><tr><td>D3</td><td>0000</td></tr><tr><td>D4</td><td>0234</td></tr><tr><td>D5</td><td>5147</td></tr><tr><td>D6</td><td>0000</td></tr><tr><td>D7</td><td>0000</td></tr></table></div></div> <div>第二次执行指令</div> 对标志位的影响：St>E，ER 为 ON	D0	0000	D1	0000	D2	507A	D3	2B68	D4	0000	D5	0234	D6	5147	D7	0000	D0	0000	D1	507A	D2	0000	D3	2B68	D4	0234	D5	0000	D6	5147	D7	0000	D0	0000	D1	5070	D2	0000	D3	2468	D4	0234	D5	0000	D6	5147	D7	0000	D0	5070	D1	0000	D2	2468	D3	0000	D4	0234	D5	5147	D6	0000	D7	0000
D0	0000																																																																		
D1	0000																																																																		
D2	507A																																																																		
D3	2B68																																																																		
D4	0000																																																																		
D5	0234																																																																		
D6	5147																																																																		
D7	0000																																																																		
D0	0000																																																																		
D1	507A																																																																		
D2	0000																																																																		
D3	2B68																																																																		
D4	0234																																																																		
D5	0000																																																																		
D6	5147																																																																		
D7	0000																																																																		
D0	0000																																																																		
D1	5070																																																																		
D2	0000																																																																		
D3	2468																																																																		
D4	0234																																																																		
D5	0000																																																																		
D6	5147																																																																		
D7	0000																																																																		
D0	5070																																																																		
D1	0000																																																																		
D2	2468																																																																		
D3	0000																																																																		
D4	0234																																																																		
D5	5147																																																																		
D6	0000																																																																		
D7	0000																																																																		

3.8.1 数据移位寄存器指令 SFT

图3.52是使用 SFT 指令的例子。SFT 指令的首通道和末通道都是 W0，说明移位是在 W0 通道内进行的。P\_1s 产生的秒脉冲作为移位脉冲，0.00 的 ON、OFF 状态作为输入数据。在 PLC 加电后的第一个扫描周期由 P\_First\_Cycle(对应内存 A200.11)对移位寄存器进行复位。在移位过程中，只要 0.01 为 ON，移位寄存器即复位。下面结合图 3.52 (c) 的工作波形说明执行 SFT 指令的移位过程。

PLC 加电之初，W0 通道内各位均为 OFF。当 0.01 为 OFF 后，在 SP 端输入的第一个移位脉冲前

沿时刻, 0.00 的 ON 状态移入 W0.00, 使 W0.00 变为 ON, W0.00 原来的 OFF 态移入 W0.01, 依此类推。在第二个移位脉冲前沿时刻, 由于 0.00 已为 OFF, 所以 W0.00 为 OFF, 而 W0.00 原来的 ON 状态移入 W0.01, 依此类推。在第四个移位脉冲前沿时刻 W0.03 变为 ON, 使 100.00 为 ON。在第五个移位脉冲时 W0.03 为 OFF, 100.00 也变为 OFF。在第六个移位脉冲到来之前 0.01 为 ON, 将 W0 通道全部复位。

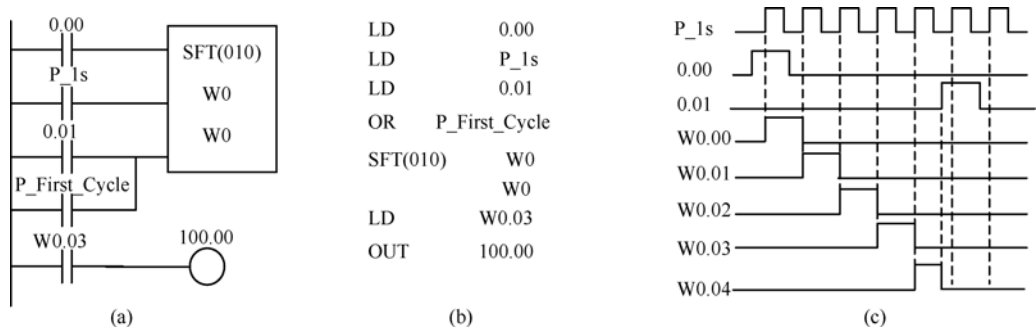


图 3.52 使用 SFT 指令的例子

在图 3.52 中, 如果将常开触点 W0.03 与 0.00 并联, 请读者自行分析移位过程。

图 3.53 是 SFT 指令的另一种用法。图中, 数据输入端接的是 P\_Off(常 OFF), 移位数据是通过@MOV 指令传送的。在 0.00 由 OFF 变为 ON 时刻执行一次@MOV 指令, 将数据#0001 传送到 W0 通道中去, 使 W0.00 为 ON, 其余位均为 OFF。对应每一个移位脉冲, W0.00 的 ON 状态依次向高位移动。该图的移位过程与图 3.52 类似, 请读者自行分析。

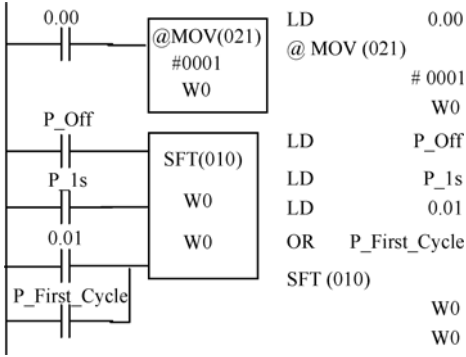


图 3.53 SFT 指令另一种用法

3.8.2 可逆移位寄存器指令 SFTR/@SFTR

图 3.54 是使用 SFTR 指令的例子。图中 0.04 是 SFTR 指令的执行条件, W10 是控制通道, H10~H11 组成可逆移位寄存器。当 0.04 为 ON 时, SFTR 指令执行移位操作; 当 0.04 为 OFF 时, SFTR 指令不执行, 此时控制通道的控制位不起作用, H10~H11 及 CY 位的数据保持不变。

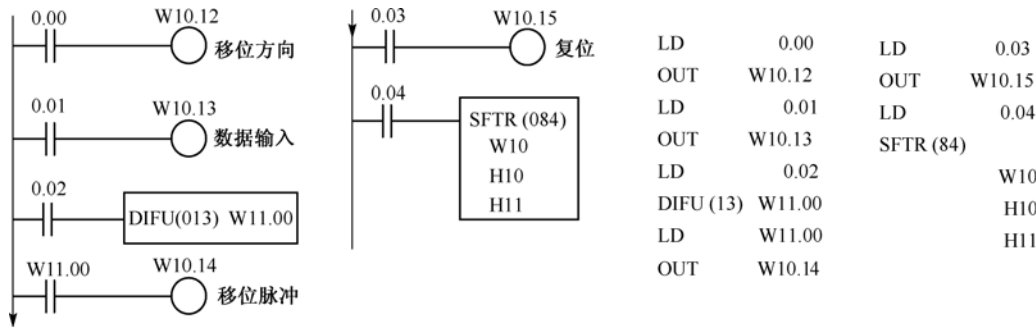


图 3.54 使用 SFTR 指令的例子

控制通道 W10 的 bit 12~bit 15 的状态是由 0.00~0.03 控制的, 其作用是:

若 0.00 为 ON，则 W10.12 为 1，执行左移位操作；若 0.00 为 OFF，则 W10.12 为 0，执行右移位操作。

若 0.01 为 ON，则 W10.13 为 1，即输入数据为 1；若 0.01 为 OFF，则 W10.13 为 0，即输入数据为 0。  
此处以 0.02 的微分信号作为移位信号。每当 0.02 由 OFF 变为 ON 时，W11.00 和 W10.14 都会 ON 一个扫描周期，由此形成移位脉冲。如果直接以 0.02 作为移位脉冲，当 0.02 为 ON 时，每个扫描周期都要执行一次移位，这将造成移位失控。

若 0.03 为 ON，则 W10.15 为 ON，可逆移位寄存器 H10~H11 及 CY 位清 0；若 0.03 为 OFF，则 W10.15 为 OFF，此时根据 W10.12 的状态将执行左移或右移操作。

当 W10.15 为 OFF、0.04 为 ON，执行左移或右移操作时，其移位过程如下：

若 W10.12 为 1，每当 0.02 由 OFF 变为 ON 时刻，H10~H11 中的数据按位依次左移一位。W10.13 的状态进入 H10.00，H11.15 的数据进入 CY 位。

若 W10.12 为 0，每当 0.02 由 OFF 变为 ON 时刻，H10~H11 中的数据按位依次右移一位。W10.13 的状态进入 H11.15，H10.00 的数据进入 CY 位。

图 3.55 中使用的是微分型指令@SFTR。在 0.04 由 OFF 变为 ON 时只执行一次移位，控制通道各控制位的状态只在一个扫描周期中有效，所以图中可以直接使用 0.02 进行移位控制。该图的工作情况与图 3.54 相同。

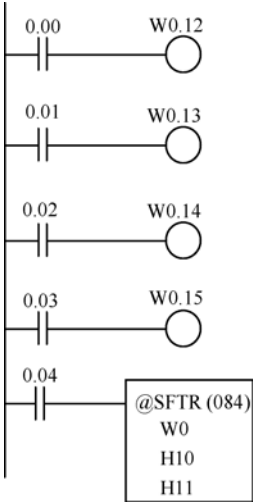


图 3.55 @SFTR 指令举例

3.8.3 数字左、右移位指令 SLD/@SLD、SRD/@SRD

图 3.56 是使用@SLD 指令的例子。图 3.56(b)是执行一次@SLD 指令后的结果通道的状态。@SLD 指令的执行条件是 0.01，其数据 St 和 E 均为 H0，说明移位是在 H0 通道内进行的。该图的功能如下：

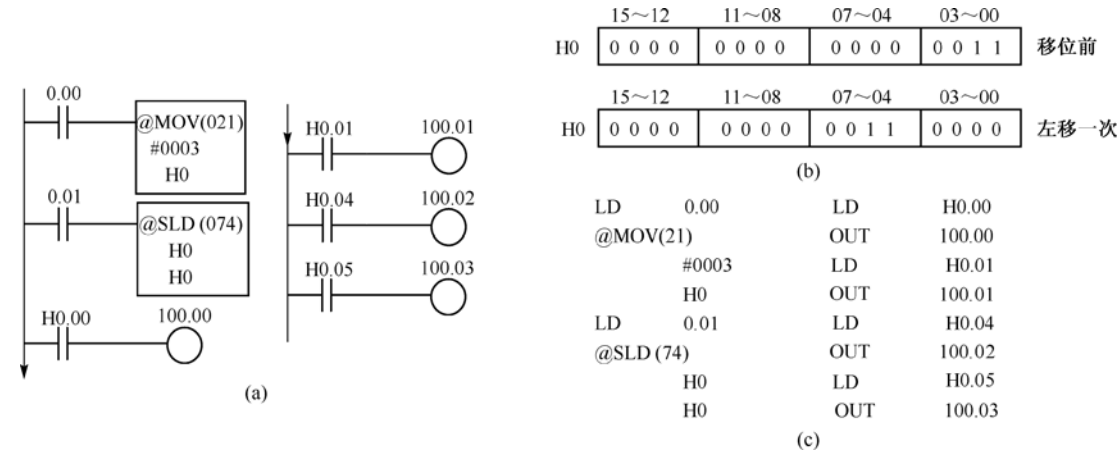


图 3.56 使用@SLD 指令的例子

当 0.00 由 OFF 变为 ON 时，执行一次 MOV 指令，将#0003 传送到 H0 通道中。由图 3.56(b) 可见，只有 H0.00 和 H0.01 为 ON，所以 100.00 和 100.01 立即变为 ON。当 0.01 由 OFF 变为 ON 时，执行一次左移位。第一次移位后，由图 3.56(b) 可见，H0.00 和 H0.01 变为 OFF，H0.04 和 H0.05 变为 ON，于是 100.00 和 100.01 变为 OFF，100.02 和 100.03 变为 ON。此后，每当 0.01 由 OFF 变为 ON 时，就执行一次左移位，利用 H0 各位的状态可以编写相应的控制程序。



3.8.4 字移位指令 WSFT/@WSFT

图3.57是使用@WSFT 指令的例子。图3.57 (b) 只画出执行一次@WSFT 指令前后 H0 和 H1 两个通道的状态。

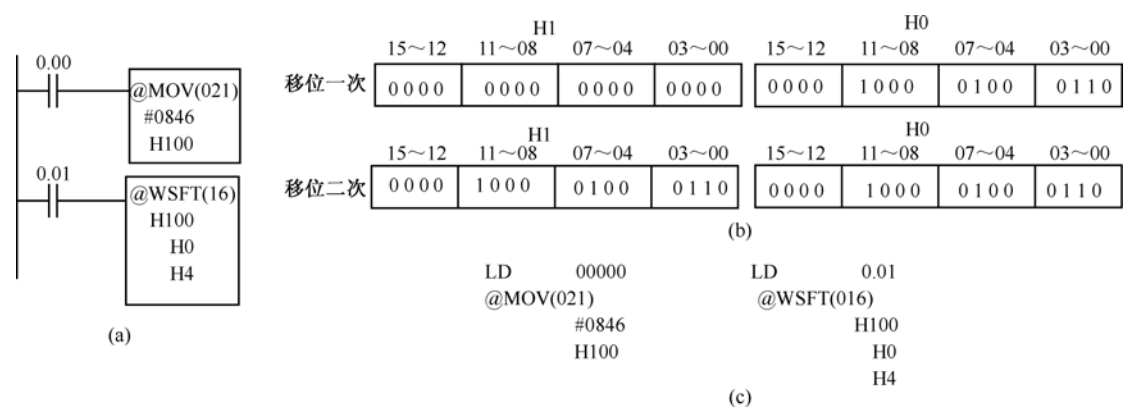


图 3.57 使用@WSFT 指令的例子

图3.57中，当 0.00 由 OFF 变为 ON 时，执行一次 MOV 指令向 H0 中传送数据#0846。每当 0.01 由 OFF 变为 ON 时，执行一次@WSFT 指令。可以推断，经过执行 5 次@WSFT 指令，#0846 就被移到 H4 中，而 H0~H4 全部为#0846。

在使用移位指令时，要根据以下几方面的需要，选择不同的移位指令：

- ① 需要位移位、数字移位，还是字移位；
- ② 需要单向移位，还是循环移位；
- ③ 是否需要标志位 CY 参与移位。

3.9 数据转换指令

表3.21 是数据转换指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。表中所列数据转换类指令都有上升沿微分功能，为简洁只列出梯形图指令的原来形式。在梯形图指令的助记符前面加@就可成为梯形图指令的微分形式。

表 3.21 数据转换指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
BCD→BIN 转换指令 BIN/@ BIN		S 是源通道(其内容为 BCD 数)，范围：CIO、W、H、T/C、A、D、@/*D、DR 等 R 是结果通道，范围与 S 的区别为 A 区变为 A448~959	当执行条件为 ON 时，将 S 中的 BCD 码转换成二进制数(S 中的内容保持不变)并存入 R 中 对标志位的影响： ① 当 S 的内容不是 BCD 码时，ER 为 ON ② 当转换结果为 0000 时，EQ 为 ON ③ N 为 OFF
BIN→BCD 转换指令 BCD/@BCD		S 是源通道(其内容为二进制数) R 是结果通道(内容为 BCD 数) S、R 的范围同上	当执行条件为 ON 时，将 S 中的二进制数转换成 BCD 码(S 中的内容保持不变)并存入 R 中 对标志位的影响： ① 当 BIN 数不在 0000H~270FH 时，ER 为 ON ② 当转换结果为 0000 时，EQ 为 ON

(续表)

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响														
4→16/8→256 译码指令 MLPX/@MLPX		<p>S 是源通道, C 是控制字; S、C 范围: CIO、W、H、T/C、A、D、@/*D、DR、#(仅 C)</p> <p>R 是结果通道, 其范围与 S 的区别为 A 区变为 A448~959</p> <p>4→16译码: S <table border="1"><tr><td>3</td><td>2</td><td>1</td><td>0</td></tr></table> C <table border="1"><tr><td>0</td><td>0</td><td></td><td></td></tr></table> 指定要译码的第一个数字(0~3) 指定要译码的数字数(0~3) 0: 1个 1: 2个 2: 3个 3: 4个</p> <p>8→256译码: S <table border="1"><tr><td>1</td><td>0</td></tr></table> C <table border="1"><tr><td>1</td><td>0</td></tr></table> 指定要译码的第一个字节(0~1) 指定要译码的字节数(0~3) 0: 1个 1: 2个</p>	3	2	1	0	0	0			1	0	1	0	<p>执行条件为 ON 时, 对 S 中指定的数字进行译码, 由 C 确定译码的方式、译码的起始数字及译码的数目</p> <p>4→16 译码: 将源通道 S 中 要译码的数字转化为十进制数 0~15, 再将结果通道中与该十进制数对应的位置为 ON, 其余位为 OFF。最多译码 4 个, 最多占用 R~R+3 4 个结果通道, 示意图如下:</p> <p>C = #0030      C = #0031      C = #0023</p> <p>C=#0002 4→16, 从数字2位开始译码, 译1位</p> <p>8→256 译码: 将源通道 S 中要译码的字节转化为十进制数 0~256, 再将结果通道中与该十进制数对应的位置 ON, 最多占用 R~R+15, R+16~R+31 二组通道</p> <p>对标志位的影响: 当 C 超出数据区范围, ER 为 ON</p>		
3	2	1	0														
0	0																
1	0																
1	0																
16→4/256→8 编码指令 DMPX/@DMPX		<p>S: 转换数据首通道 R: 结果通道 C: 控制字 S、C 范围: CIO、W、H、T/C、A、D、@/*D、DR(仅 C)、#(仅 C) R 范围与 C 的区别为 A 区变为 A448~959 C 各数字位的含义:</p> <p>16→4 编码 R <table border="1"><tr><td>3</td><td>2</td><td>1</td><td>0</td></tr></table> C <table border="1"><tr><td>0</td><td></td><td></td><td></td></tr></table> R中接受编码第一个数字号(0~3) 编码结果数字数 0~3Hex: 1~4个 编码取值 0/1: ON最高位/最低位</p> <p>256→8编码: 源通道最多: S~S+31 (2个256) R <table border="1"><tr><td>1</td><td>0</td></tr></table> C <table border="1"><tr><td>0</td><td></td><td></td><td></td></tr></table> R中接受编码第一个数字号(0~1) 编码结果数字数 0~1Hex: 1~2个 编码取值 0/1: ON最高位/最低位</p>	3	2	1	0	0				1	0	0				<p>执行条件为 ON 时, 对 S 中指定的数字进行编码, 由 C 确定编码的方式、编码结果通道的首字位及编码的通道数目</p> <p>16→4 编码: 将被编码的最多 4 个通道中为 ON 的最高/低位的位号, 分别编为一位十六进制数, 再将结果送到结果通道指定的数字位, 示意图如下:</p> <p>C=#0002: 16→4, 一个通道, 存在R的第2位</p> <p>256→8 编码: 将最多两组连续 16 个通道 256 位通道中为 ON 的最高/低位位号, 分别编为一个字节的十六进制数, 再分别送到结果通道指定的字节, 如下图:</p> <p>C=#1011: 256→8</p> <p>ER 为 ON:</p> <ul style="list-style-type: none"><li>① 当 C 超出数据区范围</li><li>② 源通道中没有为 1 的位</li></ul>
3	2	1	0														
0																	
1	0																
0																	

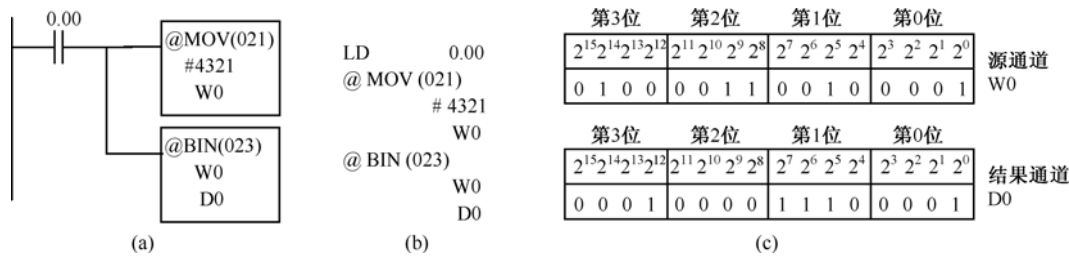
(续表)

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
ASCII 码转换指令 ASC/@ ASC		<p>S: 转换数据首通道 R: 结果通道 C: 控制字 S、C 范围: CIO、W、H、T/C、A、D、@/*D、DR、#(仅 C) R 范围与 S 的区别为 A 区变为 A448~959 C 各数字位的含义:</p> <p>S 中第一个转换的数字编号(0~3) 转换数字 0~3 (1~4 个) 转换结果输出位置 0: 低 8 位 1: 高 8 位 校验位: 0/1/2: 无校验/偶校验/奇校验</p>	<p>执行条件为 ON 时, 根据控制数据 C, 将 S 中指定的数字转换成 ASCII 码, 并存在从 R 开始的结果通道中, 见下图:</p> <p>C=# 0013</p> <p>C=# 0113</p> <p>C=# 0030</p> <p>C=# 0130</p> <p>结果通道低 8 位的 bit00~bit06 和高 8 位的 bit08~bit14 存放结果。bit07 和 bit15 是校验位。若 C 指定不校验, 则校验位为 0; 若为偶校验, 则校验位与 ASCII 码中 1 的总数应是偶数; 若为奇校验, 则校验位与 ASCII 码中 1 的总数应是奇数</p> <p>对标志位的影响同上</p>
七段解码器 指令 SDEC/@SDEC		<p>S 是源通道(内容为 BCD 码) R: 结果通道 C 是控制数据 S 范围: CIO、W、H、T/C、A、D、@/*D、DR、#(仅 C) C 各数字位的含义:</p> <p>S 中第一个要译码的数字位(0~3) S 中要译码的数字位的位数: 0: 1 位, 1: 2 位 2: 3 位, 3: 4 位 指定从 R 的高 8 位还是从低 8 位开始存放第一个转换结果 0: 低 8 位, 1: 高 8 位 固定为 0</p>	<p>当执行条件为 ON 时, 对 S 中的数字进行译码, (由 C 确定要译码的起始数字位及译码的位数)。译码结果存放在 R 中(由 C 确定是从 R 的低 8 位还是高 8 位开始存放)。R 中的 bit07 和 bit15 不用, bit00~bit06 及 bit08~bit14 分别对应数码管的 a、b、c、d、e、f、g 段。转换过程如下: C=#0003 只译数字位, 存在 R 的低 8 位</p> <p>源通道: 0001 1100 0111 0011 结果通道: ×000 0110 gfedcba gfedcba 数字位 3 是 1 转换结果</p> <p>译码结果在 R 中存放的顺序示意如下图所示:</p> <p>#0030</p> <p>#0133</p> <p>对标志位的影响同上</p>

3.9.1 BCD 码→BIN 码转换指令 BIN/@BIN

图3.58是使用@BIN 指令的例子。图3.58(c)是转换后源通道与结果通道的内容。图中, 当 0.00 由 OFF 变 ON 时, 执行一次@MOV 指令将 BCD 码#4321 传送到源通道 W0 中; 执行一次@BIN 指令将 W0 中的 BCD 码转换成二进制数, 并存放在结果通道 D0 中。转换前、后源通道 W0 的内容不变。

转换的原理是: 4 位 BCD 码可以分解成若干个  $2^n$  的十进制数的和, 4321 可分解为:  $4321=4096+128+64+32+1=2^{12}+2^7+2^6+2^5+2^0$ 。因此, 结果通道 D0 中的 bit12、bit07、bit06、bit05、bit00 应为 1。



第3位	第2位	第1位	第0位
$2^{15}2^{14}2^{13}2^{12}$	$2^{11}2^{10}2^92^8$	$2^72^62^52^4$	$2^32^22^12^0$
0 1 0 0	0 0 1 1	0 0 1 0	0 0 0 1

源通道  
W0

第3位	第2位	第1位	第0位
$2^{15}2^{14}2^{13}2^{12}$	$2^{11}2^{10}2^92^8$	$2^72^62^52^4$	$2^32^22^12^0$
0 0 0 1	0 0 0 0	1 1 1 0	0 0 0 1

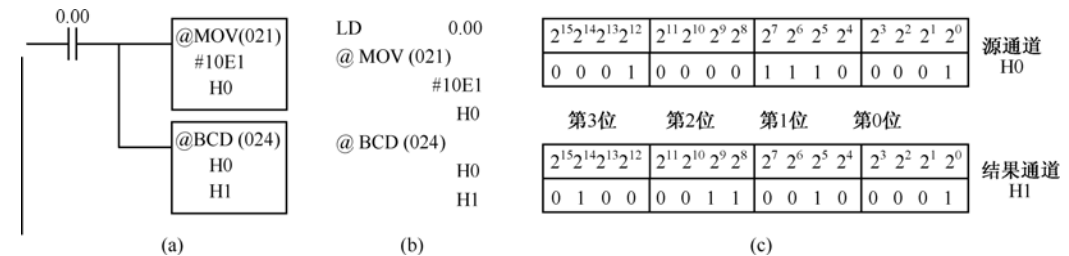
结果通道  
D0

(c)

图 3.58 使用@BIN 指令的例子

3.9.2 BIN 码→BCD 码转换指令 BCD/@BCD

图3.59是使用@BCD 指令的例子。二进制数 0001 0000 1110 0001 用十六进制数表示为 10E1。图中，当 0.00 为 ON 时，执行一次@MOV 指令，将 10E1 传送到源通道 H0 中，执行一次@BCD 指令后将 H0 中的二进制数转换成 BCD 码，并存放在结果通道 H1 中。图3.59(c)是转换后源通道与结果通道的内容。



LD 0.00

@ MOV (021)

#10E1

H0

@ BCD (024)

H0

H1

(b)

第3位	第2位	第1位	第0位
$2^{15}2^{14}2^{13}2^{12}$	$2^{11}2^{10}2^92^8$	$2^72^62^52^4$	$2^32^22^12^0$
0 0 0 1	0 0 0 0	1 1 1 0	0 0 0 1

源通道  
H0

第3位	第2位	第1位	第0位
$2^{15}2^{14}2^{13}2^{12}$	$2^{11}2^{10}2^92^8$	$2^72^62^52^4$	$2^32^22^12^0$
0 1 0 0	0 0 1 1	0 0 1 0	0 0 0 1

结果通道  
H1

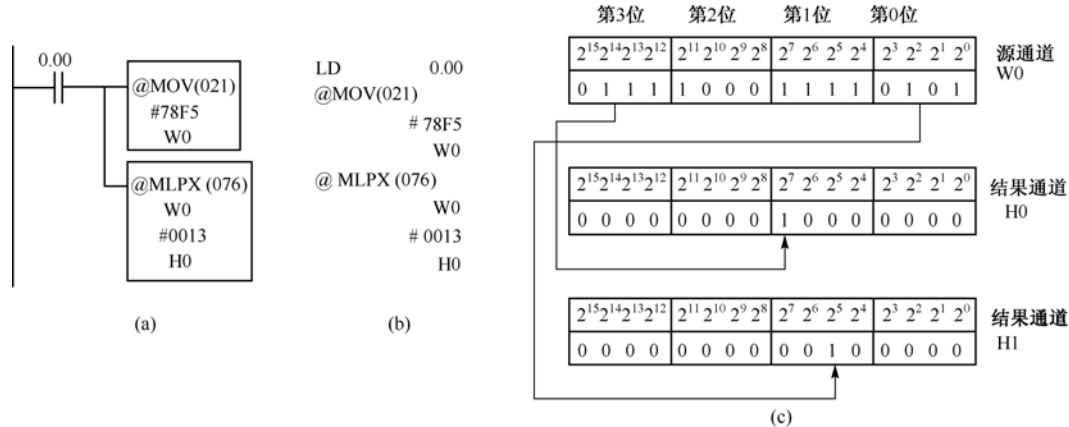
(c)

图 3.59 使用@BCD 指令的例子

转换的原理是：二进制数 0001 0000 1110 0001 对应的十进制数为  $2^{12}+2^7+2^6+2^5+2^0=4321$ 。将 4321 用 BCD 码表示，因此，转换后结果通道 H1 中的各数字位从高到低依次为 0100 、0011、0010、0001，如图3.59(c)所示。

3.9.3 4→16/8→256 译码指令 MLPX

图3.60是使用@MLPX 指令的例子。图中，译码指令的源道号为 W0(内容为 78F5)，H0 是结果通道的首通道号。控制字 C=#0013，表明要对源通道中的两个数字进行译码，从源通道的第 3 位数字开始译码，译码的顺序为，第 3 位→第 0 位。对两个数字译码的结果需要两个通道来存放，本例中结果通道是 H0 和 H1。



LD 0.00

@MOV(021)

# 78F5

W0

@ MLPX (076)

W0

# 0013

H0

(b)

第3位	第2位	第1位	第0位
$2^{15}2^{14}2^{13}2^{12}$	$2^{11}2^{10}2^92^8$	$2^72^62^52^4$	$2^32^22^12^0$
0 1 1 1	1 0 0 0	1 1 1 1	0 1 0 1

源通道  
W0

第3位	第2位	第1位	第0位
$2^{15}2^{14}2^{13}2^{12}$	$2^{11}2^{10}2^92^8$	$2^72^62^52^4$	$2^32^22^12^0$
0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0

结果通道  
H0

第3位	第2位	第1位	第0位
$2^{15}2^{14}2^{13}2^{12}$	$2^{11}2^{10}2^92^8$	$2^72^62^52^4$	$2^32^22^12^0$
0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0

结果通道  
H1

(c)

图 3.60 使用@MLPX 指令的例子

译码的原理是：源通道的第3位数字是0111，译码为十进制数的7，则以7为位号，将结果首通道H0中的bit07置为1；源通道的第0位数字是0101，译码为十进制数的5，则以5为位号，将H1的bit05置为1。图3.60(c)是转换后源通道与结果通道的内容。

对本例，如果控制字C=#0023，即对3个数字进行译码的话，第3个被译码的数字是源通道的第1位数字，即紧挨着前一个被译码的数字位，第3个译码结果应该存放在下一个结果通道H2中，即紧挨着存放前一个译码结果的通道，依此类推。

综上所述，使用译码指令MLPX时要注意：

① R+3 不能超过其所在区域。例如，在C=#0023时，若结果首通道号设为H510就会出错。因为进行3位数字译码时需要3个结果通道，而H区的最大通道号为511。

② 要牢记控制字的内容及意义，注意控制字的内容不能写错，若C设置错误，程序将无法执行。例如，C被设置成#0042，即对源通道的5位数字进行译码，而源通道不存在第5位数字。

### 3.9.4 16→4/256→8 编码指令 DMPX

图3.61是使用@DMPX指令的例子。图中，编码指令的首源通道为H0，结果通道为D0，以W20中的内容(0013)作为控制字，表明进行16→4进制编码，要对两个源通道进行编码，从结果通道的第3位数字位开始存放结果。对两个源通道进行编码有两个结果，所以只占用结果通道的两个数字位。

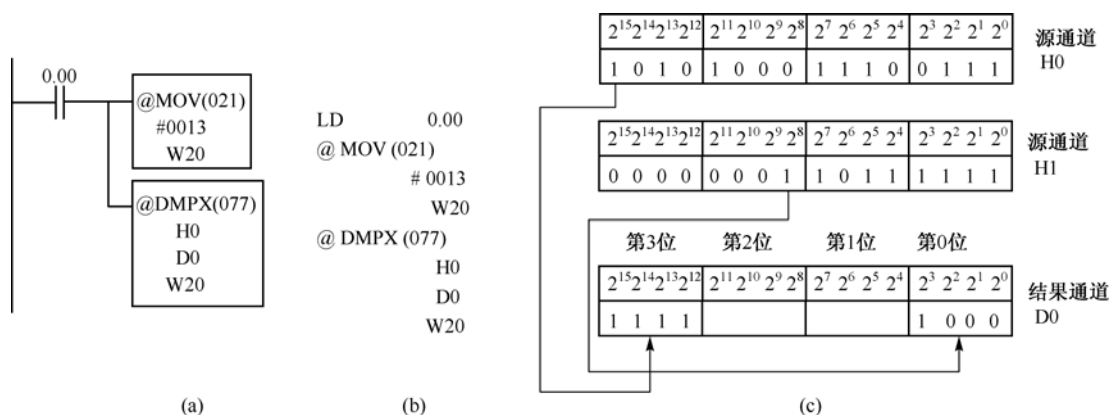


图 3.61 使用@DMPX 指令的例子

设H0中的内容为A8E7，H1中的内容为01BF。编码的原理是：首通道H0的内容A8E7(1010 1000 1110 0111)，为1的最高位的位号是15，其十六进制编码为F，这个结果要放在结果通道D0的第3位数字位；H1的内容01BF(0000 0001 1011 1111)，为1的最高位的位号是8，其十六进制编码为8，这个结果要放在结果通道D0的第0位数字位中。图3.61(c)是转换后源通道与结果通道的内容。

对图3.61，如果W20中的内容为0023，即对3个源通道进行编码，第3个被编码的源通道应该是H2，即紧挨着前一个被编码的通道，第3个编码结果应该存放在结果通道D0的第1位数字位，即紧挨着前一个编码结果的存放位，依此类推。

综上所述，使用编码指令@DMPX时要注意：

① S+3 不能超过其所在区域。例如，在控制字C=#0023时，若源首通道设为H510就会出错。因为要对3个通道进行编码，而H区只有H510和H511这两个通道可以供编码了。

② 要牢记控制字的内容及意义，并注意控制字的内容不能写错。由于一个结果通道只能存放4个转换结果，所以一次只能对4个源通道进行编码。若C设置错误，程序将无法执行。例如，若C设为#0042，即对5个源通道进行编码，而5个编码结果，一个结果通道是无法存放的。

3.9.5 ASCII 码转换指令 ASC

图3.62是执行 ASC 指令的几种情况。图 3.62(a)中 C=#0011, 表示从源通道 S 的数字位 1 开始转换, 转换 2 位, 转换结果从结果通道 R 的低 8 位开始存放, 不校验。

源通道 S 的数字位 1 是 BCD 数 3, 转换成 ASCII 码是 33, 数字位 2 是 BCD 数 1, 转换成 ASCII 码是 31。由于 C 指定不校验, 所以 bit07 和 bit15 都写 0。

图3.62(b)中 C=#1010, 示意从源通道 S 的数字位 0 开始转换, 转换 2 位, 转换结果从结果通道 R 的低 8 位开始存放, 偶校验。

源通道 S 的数字位 0 和 1 是 BCD 数 3 和 1, 转换成 ASCII 码是 33 和 31, 由于 C 指定为偶校验, 在 R 的低 8 位中, ASCII 码里 1 的个数已是偶数, 所以 bit07 写 0; 在 R 的高 8 位中, ASCII 码里 1 的个数不是偶数, 所以 bit15 写 1。

图3.62(c)中 C=#2010, 指定为奇校验, 在 R 的低 8 位中, ASCII 码里 1 的个数不是奇数, 所以 bit07 写 1; 在 R 的高 8 位中, ASCII 码里 1 的个数已是奇数, 所以 bit15 写 0。

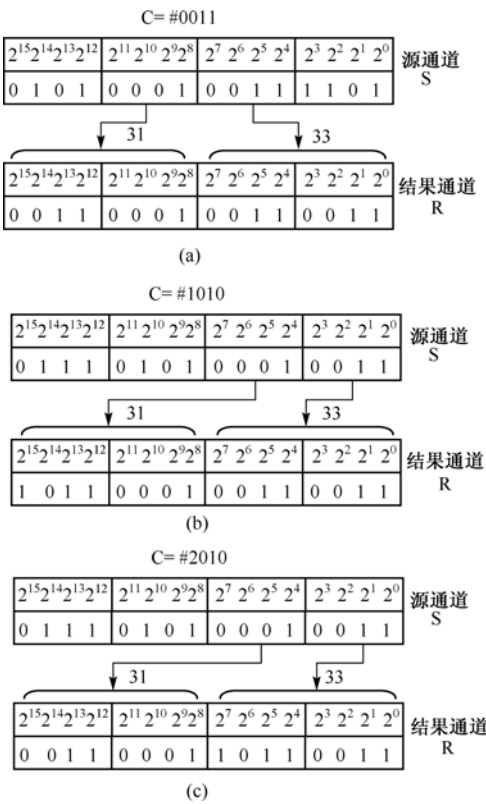


图 3.62 执行 ASC 指令的几种情况

3.9.6 七段译码指令 SDEC

图3.63是使用七段译码指令@SDEC 的例子。图 3.63(c)是译码后源通道与结果通道的内容,图 3.63(d)是七段数码管各段与结果通道各位的对应关系,图 3.63(e)是译码后 H1 中第 1 位数字和第 3 位数字所对应的数码管显示的数字。

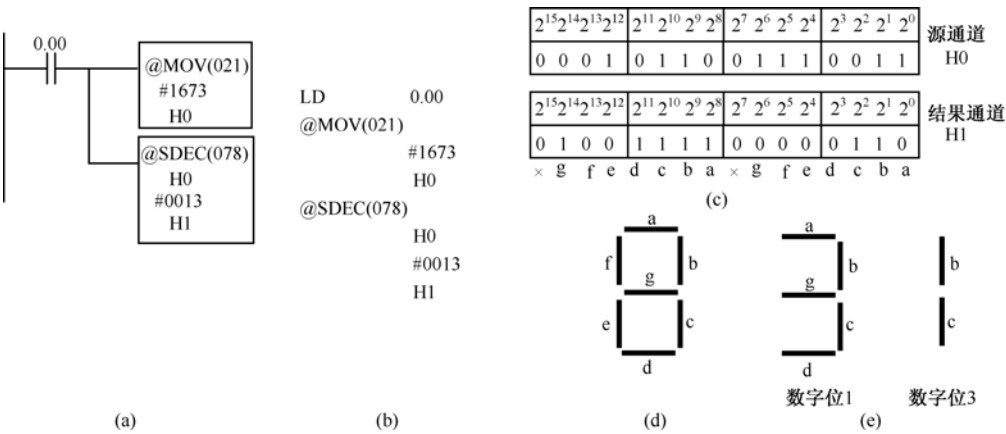


图 3.63 使用 @SDEC 指令的例子

图3.63中, 指令的源通道号为 H0, 结果通道为 H1。控制字 C=#0013, 表明从源通道的第 3 位数字开始, 对两个数字进行译码, 译码的顺序为第 3 位数字→第 0 位数字; 从结果通道的低 8 位开始接受第一个转换结果, 每个结果占 8 位, 所以只占用一个结果通道。

译码的原理是：当 0.00 为 ON 时，执行@SDEC 指令对 H0 中的数据(为 1673)进行七段译码。源通道中的第 3 位数字是 0001，经过七段译码后，七段数码管应该显示数字 1，即七段数码的 b、c 段应该是 1。第一个译码结果要存放在结果通道的低 8 位，所以 H1 的低 8 位是 0000 0110(bit7 固定为 0)；源通道中的第 0 位数字的内容为 0011，经过七段译码后，七段数码管应该显示数字 3，即七段数码的 a、b、c、d、g 段应该是 1。第二个译码结果要存放在结果通道的高 8 位，所以 H1 的高 8 位是 0100 1111 (bit15 固定为 0)。

本例中，若 C=#0113 时，虽然也是对两个数字进行译码，但需要两个结果通道。这时结果通道应是以 H1 为首通道的两个连续通道。第一个译码结果存放在 H1 的高 8 位，第二个译码结果存放在 H2 的低 8 位。

执行七段译码指令 SDEC 时，若源通道的内容有数码 A~F，七段数码管也可以显示出数码 A~F。综上所述，使用 SDEC 指令要注意：

- ① 结果通道不能超过其所在区域，例如，在控制字 C=#0113 时，若结果通道数据为 H511 就会出错。因为第一个结果要存放在 H511 的高 8 位，显然第二个译码结果无处存放。
- ② 一次最多只能对 4 个数字进行译码。若 C 设置错误，程序将无法执行。例如，C 设置为#0042，即对 5 个数字进行译码，这显然是错误的。

3.10 递增/递减指令

递增和递减指令主要实现数据的累加或递减，都具有上升沿微分指令的特性。指令码为 590~597。表 3.22 是递增和递减指令的名称、梯形图符号、操作数的范围及含义、指令功能及执行指令对标志位的影响。表中所列递增/递减类指令都有上升沿微分功能，为简洁只列出梯形图指令的原来形式。在梯形图指令的助记符前面加@就可成为梯形图指令的微分形式。

表 3.22 递增/递减类指令

名 称	梯形图符号	操作数的范围及含义	指令功能及执行指令对标志位的影响
二进制递增指令 ++(590)/@( ++(590)		CH 是源数据的通道号 范围: CIO、W、H、A448~959(单字)、A448~958(双字)、T/C、D、@D、*D、DR(单字)等	执行条件为 ON 时，通道 CH(或 CH+1、CH)中的二进制数据按二进制递增 1 对标志位的影响： ① 通道内容为 0000 时 EQ 为 ON ② 有进位时 CY 为 ON ③ 结果最高位为 1 时 N 为 ON
二进制双字递增指令 ++L(591)/@ ++L(591)			
二进制递减指令 --(592)/@ --(592)			执行条件为 ON 时，通道 CH(或 CH+1、CH)中的二进制数据按二进制递减 1 对标志位的影响： ① 通道内容为 0000 时 EQ 为 ON ② 有借位时 CY 为 ON ③ 结果最高位为 1 时 N 为 ON
二进制双字递减指令 --L(593)/@ --L(593)			
十进制递增指令 ++B(594)/@ ++B(594)		CH 是源数据的通道号 范围: CIO、W、H、A448~959(单字)、A448~958(双字)、T/C、D、@D、*D、DR(单字)等	执行条件为 ON 时，通道 CH(或 CH+1、CH)中的 BCD 数按十进制递增 1 对标志位的影响： ① Ch 数据不为 BCD 数时，ER 为 ON ② 通道内容为 0000 时 EQ 为 ON ③ 有进位时 CY 为 ON
十进制双字递增指令 ++BL/@ ++BL			
十进制递减指令 --B(596)/@ --B(596)			执行条件为 ON 时，通道 CH(CH+1、CH)中的 BCD 数按十进制递减 1 对标志位的影响： ① Ch 数据不为 BCD 数时，ER 为 ON ② 通道内容为 0000 时 EQ 为 ON ③ 有借位时 CY 为 ON
十进制双字递减指令 --BL(597)/@ --BL(597)			

递增/递减类指令非常简单明了，无须单独举例说明，由于在 CX-P 编程软件中，递增/递减类指令单列一类，故单列一节。

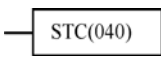
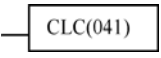
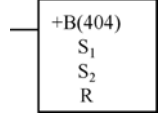
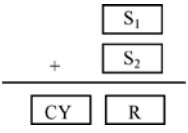
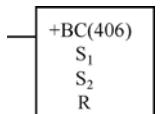
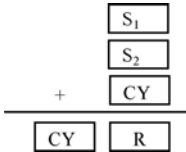
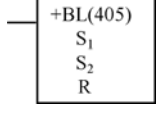
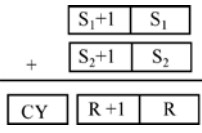
3.11 数据运算指令

CP1H 系列 PLC 提供了多种数据运算指令，按进制分为十进制数和二进制数的加、减、乘、除运算；按是否带符号分为无符号数和带符号数的加、减、乘、除运算；另外还有逻辑运算等。由于进行加、减运算时进位位也参与运算，所以对进位位置 1 和置 0 的指令 STC 和 CLC 也放在本节介绍。而在 CX-P 编程软件中 STC 和 CLC 归于特殊指令类，编程时应该从 CX-P 编程软件特殊指令类中查找。

3.11.1 十进制运算指令

表 3.23 是十进制运算指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。表中指令均具有上升沿微分功能，为简洁只列出梯形图指令的原来形式。

表 3.23 十进制数据运算指令

格 式	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
进位置 1 指令 STC/@STC		无操作数	当执行条件为 ON 时，将进位标志位 CY 置 1
进位置 0 指令 CLC/@CLC		无操作数	当执行条件为 ON 时，将进位标志位 CY 置 0
无 CY BCD 加法 运算指令 +B/@+B		S <sub>1</sub> : 被加数 (BCD) S <sub>2</sub> : 加数 (BCD) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、 A、D、@/*D、DR、#等 R: 结果通道, 范围: CIO、W、 H、T/C、A448~959、D、@/*D、 DR 等	执行条件为 ON 时, 将被加数、加数相加, 把结果存在 R 中。若结果大于 9999, 则 CY 位置 1  对标志位的影响: ① S <sub>1</sub> 或 S <sub>2</sub> 内容不为 BCD 数时, ER 为 ON ② 运算结果有进位时, CY 为 ON ③ 结果通道的内容为 0 时, EQ 为 ON
带 CY BCD 加法 运算指令 +BC/@+BC		同上	执行条件为 ON 时, 将被加数、加数及 CY 位相加, 把结果存在 R 中。若结果大于 9999, 则 CY 位置 1  对标志位的影响同上
无 CY BCD 双字 加法运算指令 +BL/@+BL		S <sub>1</sub> : 被加数首通道 (BCD) S <sub>2</sub> : 加数首通道 (BCD) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、 A、D、@/*D、#等 R: 结果通道, 范围: CIO、W、H、 T/C、A448~958、D、@/*D 等	执行条件为 ON 时, 将被加数 S <sub>1</sub> +1、S <sub>1</sub> 与加数 S <sub>2</sub> +1、S <sub>2</sub> 相加, 把结果存在 R 中。若结果大于 99999999, 则 CY 位置 1  对标志位的影响同上



(续表)

格 式	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
有 CY BCD 双字 加法运算指令 +BCL/@+BCL	<div><div>+BCL(407)</div><div>S<sub>1</sub> S<sub>2</sub> R</div></div>	同上	<p>执行条件为 ON 时, 将被加数 S<sub>1</sub>+1、S<sub>1</sub>, 加数 S<sub>2</sub>+1、S<sub>2</sub> 及 CY 位相加, 把结果存在 R 中。若结果大于 99999999, 则 CY 位置 1</p> <div><div>S<sub>1</sub>+1</div><div>S<sub>1</sub></div><div>S<sub>2</sub>+1</div><div>S<sub>2</sub></div><div>CY</div><div>+</div><div>CY</div><div>R+1</div><div>R</div></div> <p>对标志位的影响同上</p>
无 CY BCD 减法 运算指令 -B/@-B	<div><div>-B(414)</div><div>S<sub>1</sub> S<sub>2</sub> R</div></div>	<p>S<sub>1</sub>: 被减数(BCD) S<sub>2</sub>: 减数(BCD) S<sub>1</sub>、S<sub>2</sub> 范围: CIO、W、H、T/C、 A、D、@/*D、DR、#等 R: 结果通道, 范围: CIO、W、 H、T/C、A448~959、D、@/*D、 DR 等</p>	<p>执行条件为 ON 时, 将被减数、减数相减, 把结果存在 R 中</p> <p>若有借位, 则 CY 位置 1, 此时 R 中的内容为结果的十进制补码。欲得到正确的结果, 用 0 减去 R 的内容, 并将结果存在 R 中</p> <p>对标志位的影响:</p> <div><div>S<sub>1</sub></div><div>S<sub>2</sub></div><div>-</div><div>CY</div><div>R</div></div> <p>① S<sub>1</sub> 或 S<sub>2</sub> 的内容不为 BCD 数时, ER 为 ON ② 运算结果有借位时, CY 为 ON ③ 结果通道的内容为 0 时, EQ 为 ON</p>
带 CY BCD 减法 运算指令 -B/@-BC	<div><div>-BC(416)</div><div>S<sub>1</sub> S<sub>2</sub> R</div></div>	同上	<p>执行条件为 ON 时, 将被减数减去减数、再减去 CY 位, 把结果存在 R 中。若有借位, 则 CY 位置 1, 此时 R 中的内容为结果的十进制补码。欲得到正确的结果, 应先清 CY 位, 再用 0 减 R 及 CY 的内容, 并将结果存在 R 中</p> <div><div>S<sub>1</sub></div><div>S<sub>2</sub></div><div>-</div><div>CY</div><div>R</div></div> <p>对标志位的影响同上</p>
无 CY BCD 双字 减法运算指令 -BL/@-BL	<div><div>-BL(415)</div><div>S<sub>1</sub> S<sub>2</sub> R</div></div>	<p>S<sub>1</sub>: 被减数首通道(BCD) S<sub>2</sub>: 减数首通道(BCD) S<sub>1</sub>、S<sub>2</sub> 范围: CIO、W、H、T/C、 A、D、@/*D、#等 R: 结果首通道, 范围: CIO、 W、H、T/C、A448~958、D、 @/*D 等</p>	<p>执行条件为 ON 时, 将被减数 S<sub>1</sub>+1、S<sub>1</sub> 与减数 S<sub>2</sub>+1、S<sub>2</sub> 相减, 把结果存在 R 中</p> <div><div>S<sub>1</sub>+1</div><div>S<sub>1</sub></div><div>S<sub>2</sub>+1</div><div>S<sub>2</sub></div><div>-</div><div>CY</div><div>R+1</div><div>R</div></div> <p>对标志位的影响同上</p>
带 CY BCD 双字 减法运算指令 -BCL/@-BCL	<div><div>-BCL(417)</div><div>S<sub>1</sub> S<sub>2</sub> R</div></div>	同上	<p>执行条件为 ON 时, 将被减数 S<sub>1</sub>+1、S<sub>1</sub>, 减去减数 S<sub>2</sub>+1、S<sub>2</sub>, 再减去 CY 位, 把结果存在 R 中</p> <p>有借位时的处理同 -BC 指令</p> <p>对标志位的影响同上</p> <div><div>S<sub>1</sub>+1</div><div>S<sub>1</sub></div><div>S<sub>2</sub>+1</div><div>S<sub>2</sub></div><div>-</div><div>CY</div><div>R+1</div><div>R</div></div>
BCD 乘法运算指 令 *B/@*B	<div><div>*B(424)</div><div>S<sub>1</sub> S<sub>2</sub> R</div></div>	<p>S<sub>1</sub>: 被乘数(BCD) S<sub>2</sub>: 乘数(BCD) S<sub>1</sub>、S<sub>2</sub> 范围: CIO、W、H、T/C、 A、D、@/*D、DR、#等 R: 结果首通道, 范围: CIO、 W、H、T/C、A448~958、D、 @/*D 等</p>	<p>执行条件为 ON 时, 将被乘数、乘数相乘, 把结果存在 R+1、R 通道中</p> <div><div>S<sub>1</sub></div><div>S<sub>2</sub></div><div>×</div><div>CY</div><div>R</div></div> <p>对标志位的影响:</p> <p>① 被乘数或乘数不为 BCD 数时, ER 为 ON ② 结果通道的内容为 0 时, EQ 为 ON</p>

(续表)

格 式	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
BCD 双字乘法运算指令 *BL/@*BL		S <sub>1</sub> : 被乘数首通道(BCD) S <sub>2</sub> : 乘数首通道(BCD) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、A、D、@/*D、#等 R: 结果首通道, 范围: CIO、W、H、T/C、A448~956、D、@/*D 等	执行条件为 ON 时, 将被乘数 S <sub>1</sub> +1、S <sub>1</sub> , 乘数 S <sub>2</sub> +1、S <sub>2</sub> 相乘, 把结果存在 R+3~R 通道中 <div><div><div>S<sub>1</sub>+1</div><div>S<sub>1</sub></div></div><div><div>S<sub>2</sub>+1</div><div>S<sub>2</sub></div></div><div>×</div><div><div>R+3</div><div>R+2</div><div>R+1</div><div>R</div></div></div> 对标志位的影响同上
BCD 除法运算指令 /B/@/B		S <sub>1</sub> : 被除数(BCD) S <sub>2</sub> : 除数(BCD) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、A、D、@/*D、DR、#等 R: 结果首通道, 范围: CIO、W、H、T/C、A448~958、D、@/*D 等	执行条件为 ON 时, 将被除数除以除数, 商存入 R 中, 余数存入 R+1 中 <div><div>S<sub>1</sub></div><div>÷</div><div><div>S<sub>2</sub></div><div>÷</div><div><div>R+1</div><div>R</div></div></div> 对标志位的影响: ① 被除数或除数不为 BCD 数时 ER 为 ON ② 除数为 0 时 ER 为 ON ③ 结果通道为 0 时, EQ 为 ON</div>
BCD 双字除法运算指令 /BL/@/BL		S <sub>1</sub> : 被除数首通道(BCD) S <sub>2</sub> : 除数首通道(BCD) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、A、D、@/*D、#等 R: 结果首通道, 范围: CIO、W、H、T/C、A448~956、D、@/*D 等	当执行条件为 ON 时, 将被除数 S <sub>1</sub> +1、S <sub>1</sub> , 除数 S <sub>2</sub> +1、S <sub>2</sub> 相除, 商存入 R(低 4 位)和 R+1(高 4 位)中, 余数存入 R+2(低 4 位)和 R+3(高 4 位)中 <div><div><div>S<sub>1</sub>+1</div><div>S<sub>1</sub></div></div><div>÷</div><div><div>S<sub>2</sub>+1</div><div>S<sub>2</sub></div></div><div>÷</div><div><div>R+3</div><div>R+2</div><div>R+1</div><div>R</div></div></div> 对标志位的影响同上

1. 十进制加法运算指令

图3.64是使用+BC 和+BCL 指令的例子, 图3.64 (c)是执行双字加运算的操作过程。为了保证运算的正确, 每次运算前都先用 CLC 指令将进位位清零。

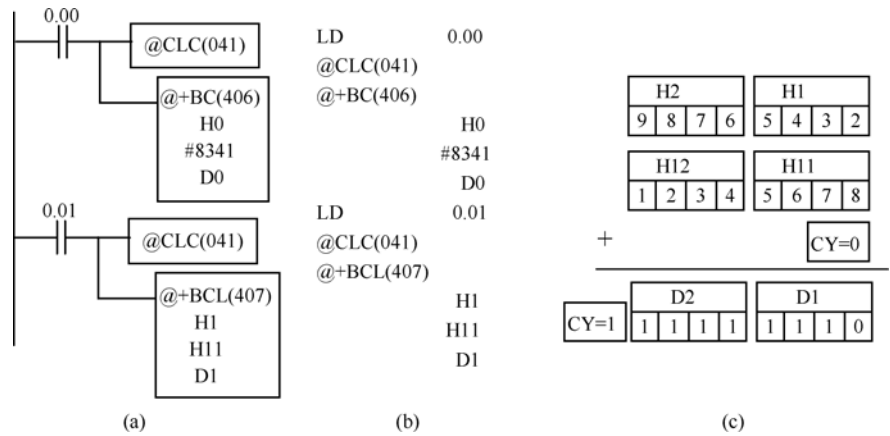


图 3.64 使用 BCD 加法指令的例子

图3.64 中, 当 0.00 为 ON 时执行@CLC 指令清进位位, 执行@+BC 指令, 将 H0(#1234)与#8341 及 CY 相加, 结果存放在 D0 中; 当 0.01 为 ON 时, 执行@CLC 指令清进位位, 执行@+BCL 指令, 将双字 H2(#9876)H1(#5432)与 H12(#1234)H11(#5678)及 CY 相加, 结果存放在 D2 和 D1 中。

图3.65是使用+BC 指令修改 TIM 设定值的例子。图中，TIM0000 的设定值是由 D10 通道提供的。程序运行前向 D10 写入初始数据#0300。这里使用+BC 指令是为了方便多次修改 TIM 的设定值。

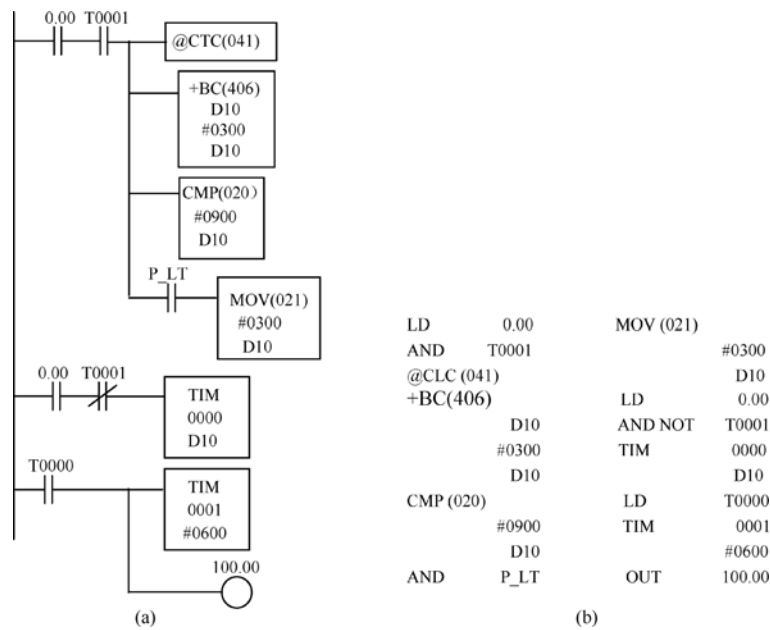
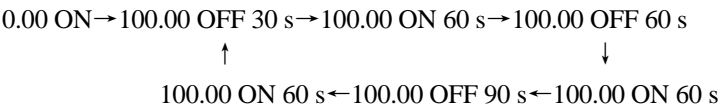


图 3.65 使用+BC 指令修改 TIM 的设定值

该段程序中,每当 T0001 为 ON(ON 一个扫描周期)时执行@+BC 指令,将 D10 中的数据加#0300,即 TIM0000 的设定值增加 30 s。当 D10 中的数据大于#0900 时,执行 CMP 指令后 P\_LT 为 ON,从而使 MOV 指令得以执行,再将#0300 传送给 D10,即令 TIM0000 的设定值恢复为 30 s。

该段程序对 100.00 实现了循环间歇 OFF、ON 的控制。100.00 每次 ON 的时间保持不变,而每次 OFF 的时间依次增加 30 s(但不超过 90 s)。0.00 对应一个自锁开关,程序实现的控制功能如下:



100.00 每次都 ON 60 s,是由 TIM0001 控制的;100.00 第一次 OFF 的时间是 30 s,以后 OFF 的时间依次增加 30 s,这是通过执行+BC 指令改变 TIM0000 的设定值实现的;100.00 OFF 的最长时间不超过 90 s,是由 CMP 指令控制的。当然,实现上述控制并不是仅此一个方案。

2. 十进制减法运算指令

图3.66是使用减法运算-BC 指令的例子。被减数在 H0 中,减数在 D0 中,结果存在 H1 中,进位位存在 H2 中。

当 0.00 为 ON 时,执行 CLC 指令清进位位。执行-BC 指令,用 H0 的内容减去 D0 的内容,再减去 P\_CY 的内容,差存在结果通道 H1 中。若运算没有借位时 P\_CY 被置 0, P\_CY 为 OFF, H2 为 0;若运算有借位时,则结果通道中的内容是差的十进制补码,故须进行第二次减法运算。由于此时 P\_CY 为 ON,于是第二次执行减法运算,结果存在 H1 中,同时把 H2 置 1。两次减法运算的操作过程如下:

	H0		D0		CY		H1		CY
第一次相减:	1000	-	2000	-	0	→	1000 + (10000 - 2000) = 9000		1
第二次相减:	0000	-	9000	-	0	→	0000 + (10000 - 9000) = 1000		1

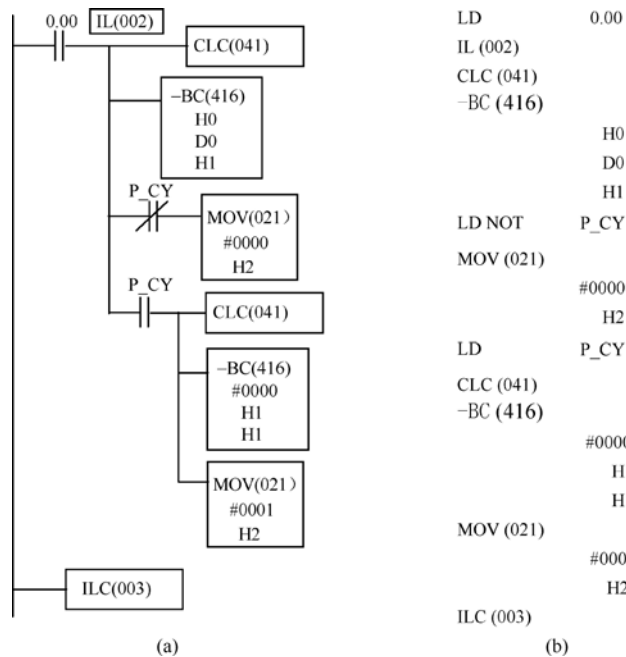


图 3.66 使用-BC 指令的例子

3. 十进制递增、递减指令及乘、除法运算指令

由于两个最大的单字 BCD 数相乘，即  $9999 \times 9999 = 99980001$ ，运算结果不发生进位，同样两个最大的双字 BCD 数相乘结果也不发生进位。所以乘、除运算都不涉及进位位 CY。

图3.67 中，使用了递增指令@++B、乘法运算指令@\*B 和除法运算指令@/B。当程序运行时先令 0.00 ON 一次，将 D0~D4 清零，为进行各种运算做好准备。

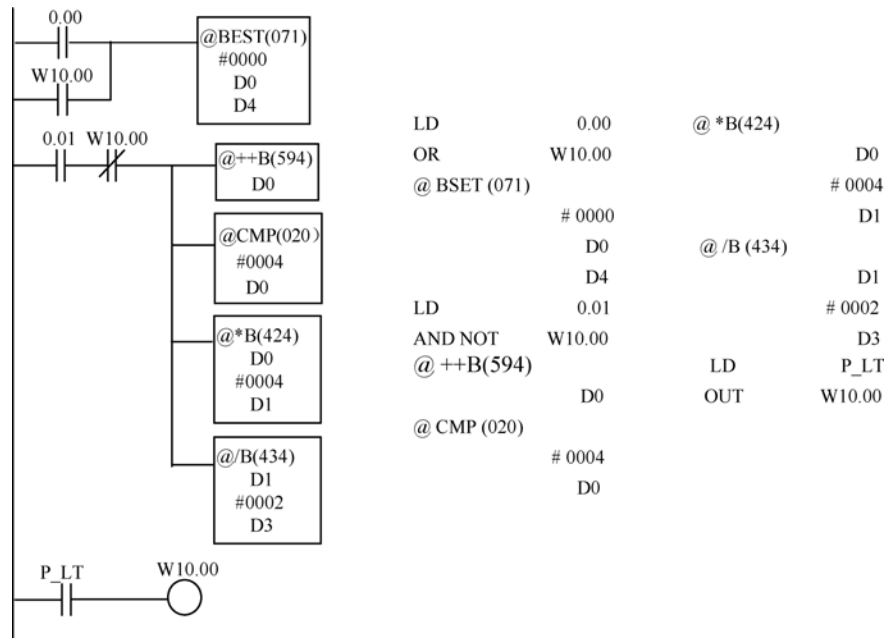


图 3.67 使用@++B、@\*B、@/B 指令的例子

每当 0.01 为 ON 时，要执行以下几个指令：执行@++B 指令，将 D0 中当前的内容加 1；执行 CMP 指令，将 D0 中的内容与#0004 比较，若 D0 的内容比#0004 大，则将 W10.00 置为 ON；执行 @\*B 指令，将 D0 中的内容与#0004 相乘，把结果存在 D1 和 D2 中；执行@/B 指令，将 D1 和 D2 中的内容与#0002 相除，把商存在 D3、余数存在 D4 中。0.01 第 1 次到第 4 次为 ON，D0~D4 的内容如下：

0.01 ON 的次数	D0	D1	D2	D3	D4
第 1 次	0001	0004	0000	0002	0000
第 2 次	0002	0008	0000	0004	0000
第 3 次	0003	0012	0000	0006	0000
第 4 次	0004	0016	0000	0008	0000

从 0.01 第 5 次 ON 开始，以后将重复上面的过程。

3.11.2 二进制运算指令

表 3.24 是 CPIH 的二进制运算指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响，包括带符号和进位位 CY 的 BIN 加减法、无符号/带符号的 BIN 乘除法，表中只列出单字二进制加、减、乘、除法指令，双字二进制加、减、乘、除法指令的用法类似，这里不一一列出。所有二进制运算指令都具有上升沿微分功能，为简洁只画出梯形图指令的原来形式。在梯形图指令的助记符前加@就可变成微分形式。

表 3.24 二进制运算指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
带符号无 CY BIN 加法运算指令 +/@+		S <sub>1</sub> : 被加数(带符号 BIN) S <sub>2</sub> : 加数(带符号 BIN) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、A、D、@/*D、DR、#等 R: 结果通道, 范围 CIO、W、H、T/CA、A448~959、D、@/*D、DR 等	执行条件为 ON 时, 将被加数、加数相加, 把结果存在 R 中。若结果大于 FFFF, 则 CY 位置 1  对标志位的影响: ① 加运算结果有进位时, CY 为 ON ② 结果通道内容为 0000 时, EQ 为 ON ③ 结果的最高位为 1 时, N 为 ON ④ 正数相加结果在 8000~FFFF 之间时, 上溢出标志 OF 为 ON ⑤ 负数相加结果在 0000~7FFF 之间时, 下溢出标志 UF 为 ON
带符号和 CY BIN 加法运算指令 +C/@+C		同上	执行条件为 ON 时, 将被加数、加数及进位位 CY 相加, 结果存在 R 中。若结果大于 FFFF, 则 CY 置 1  对标志位的影响同上

(续表)


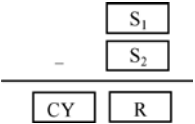

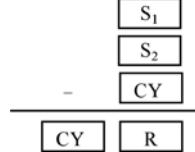

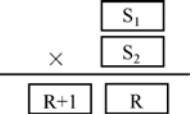
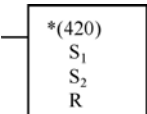
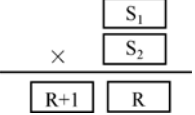


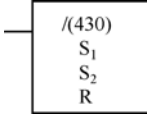
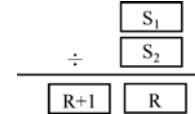
名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
带符号无 CY BIN 减法运算指令 -/@-		S <sub>1</sub> : 被减数(带符号 BIN) S <sub>2</sub> : 减数(带符号 BIN) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、A、D、@/*D、DR、#等 R: 结果通道, 范围 CIO、W、H、T/CA、A448~959、D、@/*D、DR 等	执行条件为 ON 时, 将被减数减去减数, 结果存在 R 中。有借位时, CY 为 ON, R 中的内容为结果的二进制补码, 要用 0 减去 R 的内容, 结果存在 R 中  对标志位的影响: ① 减运算结果有借位时, CY 为 ON ② 结果通道内容为 0000 时, EQ 为 ON ③ 结果通道内容最高位为 1 时, N 为 ON ④ 正数减负数结果在负数 8000~FFFF 之间时 OF 为 ON ⑤ 负数减正数结果在正数 0000~7FFF 之间时 UF 为 ON
带符号和 CY BIN 减法运算指令 -C/@-C		同上	执行条件为 ON 时, 将被减数减去减数及 CY, 结果存在 R 中。有借位时, CY 为 ON, R 中的内容为结果的二进制补码, 要先清 CY, 再用 0 减去 R 及 CY 的内容, 结果存在 R 中  对标志位的影响: ①②③同上 ④ 正数减负数及 CY 结果在 8000~FFFF 之间时 OF 为 ON ⑤ 负数减正数及 CY 结果在 0000~7FFF 之间时 UF 为 ON
无符号 BIN 乘法运算指令 *U/@*U		S <sub>1</sub> : 被乘数(BIN) S <sub>2</sub> : 乘数(BIN) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、A、D、@/*D、DR、#等 R: 结果通道, 范围 CIO、W、H、T/CA、A448~958、D、@/*D 等	执行条件为 ON 时, 将被乘数、乘数相乘, 把结果存在 R+1、R 中  对标志位的影响: ① 结果为 0000 时, EQ 为 ON ② 结果的最高位为 1 时, N 为 ON
带符号 BIN 乘法运算指令 */@*		S <sub>1</sub> : 被乘数(带符号 BIN) S <sub>2</sub> : 乘数(带符号 BIN) 其余同上	执行条件为 ON 时, 将被乘数、乘数相乘, 把结果存在 R+1、R 中  对标志位的影响同上
无符号 BIN 除法运算指令 /U/@/U		S <sub>1</sub> : 被除数(无符号 BIN) S <sub>2</sub> : 除数(无符号 BIN) S <sub>1</sub> 、S <sub>2</sub> 范围: CIO、W、H、T/C、A、D、@/*D、DR、#等 R 是结果通道, 范围 CIO、W、H、T/CA、A448~958、D、@/*D 等	执行条件为 ON 时, 将被除数除以除数, 结果存入 R(存商)和 R+1(存余数)通道  对标志位的影响: ① 除数为 0 时 ER 为 ON ② 运算结果之商为 0000 时, EQ 为 ON ③ 结果之商的最高位为 1 时, N 为 ON
带符号 BIN 除法运算指令 //@/		S <sub>1</sub> : 被除数(带符号 BIN) S <sub>2</sub> : 除数(带符号 BIN) 其余同上	执行条件为 ON 时, 将被除数除以除数, 结果存入 R(存商)和 R+1(存余数)通道  对标志位的影响其余同上

图 3.68 中是使用二进制运算指令完成 (250×8-1000)/50 运算的例子。图中，当 0.00 为 ON、0.01 为 OFF 时，执行@BSET 指令将 D0~D4 清零。当 0.01 为 ON、0.00 为 OFF 时，执行如下操作：

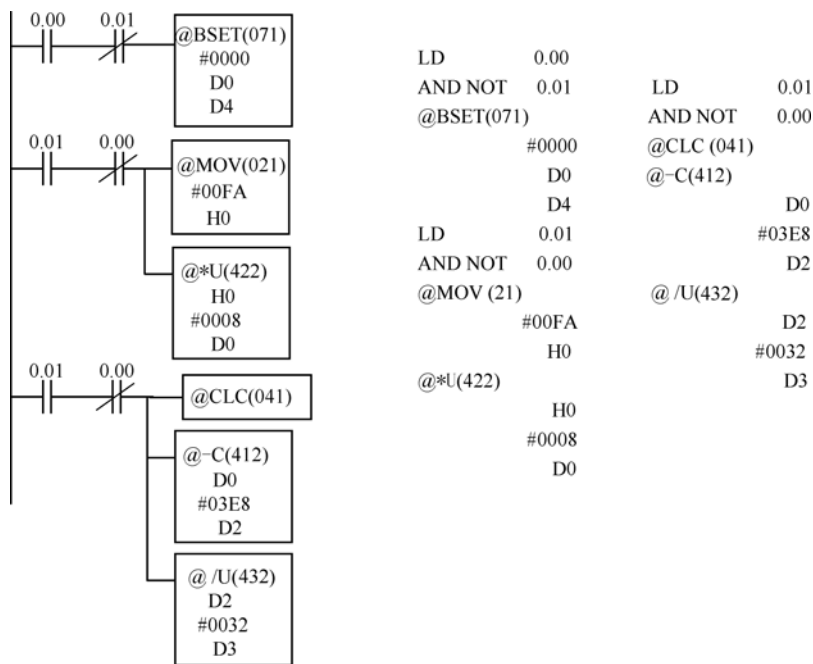


图 3.68 使用二进制运算指令的例子

执行@MOV 指令将#00FA(十进制的 250)传送到 H0 中；执行@\*U 指令将 H0 的内容与#0008 相乘，把结果的低位 07D0(十进制的 2000)存在 D0 中、结果的高位 0000 存在 D1 中；执行 @CLC 指令将 CY 清零，以准备进行相减运算。执行@-C 指令，以 D0 的内容为被减数与#03E8(十进制的 1000)相减，结果#03E8 存在 D2 中；执行@/U 指令，将 D2 中的内容除以#0032(十进制的 50)，把商#0014(十进制的 20)存在 D3 中、余数#0000 存在 D4 中。执行各种指令和运算的结果见表 3.25。

表 3.25 执行各种指令和运算的结果

执 行 指 令	H0	D0	D1	D2	D3	D4	CY
@ BSET (071)	—	0000	0000	0000	0000	0000	—
@ MOV (021)	00FA	0000	0000	0000	0000	0000	—
@*U (422)	00FA	07D0	0000	0000	0000	0000	—
@CLC (041)	00FA	07D0	0000	0000	0000	0000	0
@-C (412)	00FA	07D0	0000	03E8	0000	0000	0
@ /U (432)	00FA	07D0	0000	03E8	0014	0000	—

3.12 逻辑运算指令

表 3.26 是逻辑运算指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。表中的 5 种逻辑运算指令 COM (029)、ANDW (034)、ORW (035)、XORW (036) 和 XNRW (037) 都有相应的双字逻辑运算形式 COML (614)、ANDL (610)、ORWL (611)、XORL (612)、XNRL (613)，与单字的功能类似，这里不再赘述。所有逻辑运算指令都具有上升沿微分指令功能，为简洁只画出梯形图指令的原来形式，在梯形图指令的助记符前加@就可变成微分形式。

表 3.26 逻辑运算指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
求反指令 COM/@COM	<div>COM(029) Ch</div>	Ch 为被求反的通道号， 范围：CIO、W、H、T/C、A448~959、 D、@/*D、DR、等	执行条件为 ON 时，将通道中的数据按位求反，并 存放在原通道中 对标志位的影响： ① 结果为 0000 时，EQ 为 ON ② 结果的最高位为 1 时，N 为 ON
逻辑与运算指令 ANDW/@ANDW	<div>ANDW(034) S<sub>1</sub> S<sub>2</sub> R</div>	S <sub>1</sub> 为数据 1 S <sub>2</sub> 为数据 2 S <sub>1</sub> 、S <sub>2</sub> 范围：CIO、W、H、T/C、A、 D、@/*D、DR、#等 R 是结果通道，范围：CIO、W、H、 T/C、A448~959、D、@/*D、DR 等	执行条件为 ON 时，将输入数据 S <sub>1</sub> 和输入数据 S <sub>2</sub> 按位进行逻辑与运算，并把结果存放在通道 R 中 对标志位的影响同上
逻辑或运算指令 ORW/@ORW	<div>ORW(035) S<sub>1</sub> S<sub>2</sub> R</div>		当执行条件为 ON 时，将输入数据 S <sub>1</sub> 和输入数据 S <sub>2</sub> 按位进行逻辑或运算，并把结果存放在通道 R 中 对标志位的影响同上
逻辑异或运算指令 XORW/@XORW	<div>XORW(036) S<sub>1</sub> S<sub>2</sub> R</div>		当执行条件为 ON 时，将输入数据 S <sub>1</sub> 和输入数据 S <sub>2</sub> 按位进行逻辑异或运算，并把结果存放在通道 R 中 对标志位的影响同上
逻辑同或运算指令 XNRW/@XNRW	<div>XNRW(037) S<sub>1</sub> S<sub>2</sub> R</div>		当执行条件为 ON 时，将输入数据 S <sub>1</sub> 和输入数据 S <sub>2</sub> 按位进行逻辑同或运算，并把结果存放在通道 R 中 对标志位的影响同上

图3.69是使用逻辑指令的例子。图中，在 0.00 为 ON、0.01 为 OFF 时，执行@BSET 指令将所有存放结果的通道都清零。当 0.01 为 ON、0.00 为 OFF 时，执行如下各种逻辑运算指令：执行@ANDW 指令，将 008F 与 0081 进行逻辑“与”运算，结果 0081 放在 D0 中；执行@ORW 指令，将通道 D0 的内容与 0073 进行逻辑“或”运算，结果 00F3 放在 D1 中；执行@XORW 指令，将 D0 与 D1 两个通道的内容进行逻辑“异或”运算，结果 0072 放在 D2 中。执行各种逻辑运算的过程如图 3.70 所示。

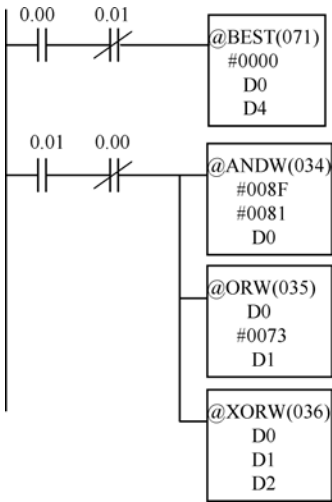


图 3.69 使用逻辑运算指令的例子

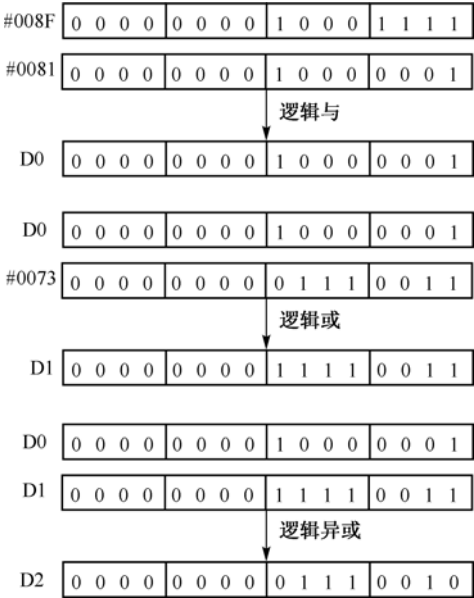


图 3.70 执行逻辑运算的过程



用逻辑指令不仅可以进行通道清零，还可以将通道中的某些位屏蔽，保留另外一些位的状态，根据欲保留和欲屏蔽位的情况设定一个常数，用 ANDW 指令将通道数据与该常数相“与”即可。例如，欲保留 H0 中的 bit0、bit3、bit4、bit7、bit10 的状态而屏蔽其余位的状态时，可以用#0499 与 H0 进行逻辑“与”来实现这个操作。

3.13 浮点转换、运算指令

OMRON 的 CQM1H、CS1、CJ1 和 CP1 等系列的 PLC 都具有浮点运算功能。本章以 CPIH 为例，介绍浮点运算指令及应用。

3.13.1 浮点数

所谓浮点数是指用符号、尾数、指数来表示实数的数据。  
在数据的位数有限的情况下，用定点数表示的数范围小。当用浮点数表示一个实数时，小数点的位置可变，随着小数点位置的改变，指数的数值也随之改变。因此，在有效数字的位数有限的情况下，不仅浮点数的范围可以比定点数的范围大得多，而且也可以保持数据的有效精度。  
对浮点数的表示法，美国电气与电子工程师协会(IEEE)制定的标准已被众多计算机制造商采用。下面以浮点数的 IEEE754 格式为例，介绍浮点数的表示法。

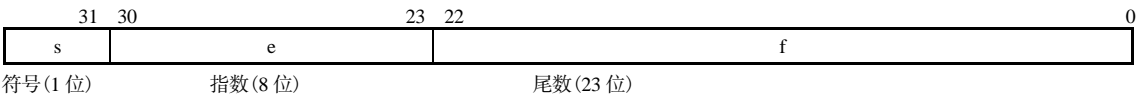
1. 浮点数的 IEEE754 格式

浮点数的 IEEE754 格式可表示如下。

实数 = (-1)<sup>s</sup> 2<sup>e-127</sup> (1.f)

式中，s 表示符号，e 表示指数，f 表示尾数(小数点后面的有效数字)。

浮点数要用 32 位表示，占两个字，各位的意义如下。



符号 s 占 1 位，为 0 表示正数，为 1 表示负数。  
指数 e 占 8 位，为 0~255，实际指数是 e 减去 127 的差，结果为-127~128。  
尾数 f 占 23 位，表示二进制浮点数的尾数部分，满足 1.0≤1.f<2.0。

2. 浮点数的有效数字位数

浮点数的有效数字为 24 位二进制数，对应的十进制数的有效数字位数为 7 位。

3. 用浮点数表示的数据范围

浮点数表示的数据范围如下：

- -∞
- -3.402823×10<sup>38</sup>~-1.175494×10<sup>-38</sup>
- 0
- 1.175494×10<sup>-38</sup>~3.402823×10<sup>38</sup>
- +∞
- NaN(Not a Number，非数：不是一个数)

4. 特殊数据 NaN、±∞、±0 的浮点数格式

- NaN: e=255, f≠0
- -∞: e=255, f=0, s=1
- +∞: e=255, f=0, s=0
- -0: e=0, f=0, s=1
- +0: e=0, f=0, s=0

5. 用浮点数表示的数据种类

用浮点数表示的数据种类见表 3.27。

表 3.27 用浮点数表示的数据种类

尾数(f)	指数(e)		
	0	非 0 和非全 1	全 1(255)
0	0	标准数据	□
非 0	非标准数据		NaN(非数)

(1) 标准数据

标准数据的浮点数表示如下。

$(-1)^s \times 2^{e-127} \times (1 + \text{尾数} \times 2^{-23})$

- 符号 s 为 0 表示正数，为 1 表示负数。
- 指数 e 为 1~254 (0 和 255 分别用于表示 0 和 ∞)，所以实际指数为 -126~127。
- 尾数 f 用 0~(2<sup>23</sup>-1) 表示，在实际尾数 1.f 中，规定位 23 为 1，二进制数的小数点要紧跟在它之后。

例如，某标准数据的浮点数如下。

31 30		23 22		1 0	
1	1 0 0 0 0 0 0 0	1 1 0			
符号	指数	尾数			

符号 s 为 1，表示该数据是负数。

指数 e 为 2<sup>7</sup>=128，实际指数为 128-127=1。

尾数 f 为 (2<sup>22</sup>+2<sup>21</sup>) × 2<sup>-23</sup>=0.75，实际尾数为 1+0.75=1.75。

该数据值为 -2<sup>1</sup>×1.75=-3.5。

必须指出，标准数据浮点数中的实际尾数 1.f 占用 24 位，规定了位 23 为 1，而且该位的 1 是隐含的。这样做的目的是去掉尾数中最高位为 1 的位之前所有的 0，以便在有效数字为 24 位二进制的前提下，尽可能提高数据的表示精度。

以十进制数 0.000123456 的表示法为例，当只能保留 6 位有效数字的时候，为了保证数据的精度，去掉数据中从左边开始第 1 个不为零的数字(本例为 1)之前所有的 0，将 0.000123456 写成 1.23456×10<sup>-4</sup>。又如，二进制数 0.00111111，当只能保留 6 位有效数字的时候，为了保证数据的精度，去掉数据中最高位为 1 的位之前所有的 0，将 0.00111111 写成 1.11111×2<sup>-11</sup>。

(2) 非标准数据

非标准数据是绝对值太小而不能表示为标准数据的实数。非标准数据的有效数字较少。若计算结果(包括中间结果)是非标准数据，则有效数字的个数将减少。

非标准数据的浮点数表示如下。

$$(-1)^s \times 2^{-126} \times (\text{尾数} \times 2^{-23})$$

- 符号 s 为 0 表示正数，为 1 表示负数。
- 指数 e 为 0，实际指数为-126。
- 尾数 f 用 1~(2<sup>23</sup>-1) 表示，在实际尾数 0.f 中，设定位 23 为 0(这也是一种规定)，二进制数的小数点要紧跟在它的后面。

例如，某非标准数据的浮点数如下。

31	30	~	23	22	~	1	0
1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
符号		指数				尾数	

符号 s 为 1，表示该数是负数。  
指数 e 为 0，实际指数为-126。  
尾数 f 为 (2<sup>22</sup>+2<sup>21</sup>) × 2<sup>-23</sup>=0.75，实际尾数为 0+0.75=0.75。  
该数据值为-2<sup>-126</sup>×0.75，是一个绝对值很小的数值。

(3) 零

+0.0 和-0.0 可通过设置符号来表示，0 表示正数，1 表示负数。指数和尾数都为 0。+0.0 和-0.0 都等于 0.0。

(4) 无穷大

+∞和-∞可通过设置符号来表示，0 表示正数，1 表示负数。指数为 255(2<sup>8</sup>-1)，尾数为 0。

(5) NaN(非数：不是一个数)

非法计算将导致 NaN(不是一个数)，例如，0.0/0.0、(+∞) + (-∞)、(+∞) - (+∞)、(-∞) - (-∞)、∞/∞或 0×∞的计算结果，不是一个数或无穷大。NaN 的指数为 255(2<sup>8</sup>-1)，尾数为非 0。

NaN 的符号或尾数(除了非 0 外)没有特别的规定。

6. 浮点数运算结果的四舍五入

当浮点数运算结果中的数字个数超过内部处理表达式中的有效数字时，若结果近似于两个内部浮点表达式中的一个，则使用较接近的值。若结果介于两个内部浮点表达式的中间，可四舍五入，以便使尾数的最后一个数字为 0。

7. 处理特殊数据注意事项

特殊数据包括 0、∞、NaN，处理时应注意以下事项。

- ① +0 和-0 的和为+0。
- ② 相同符号的 0 之差为+0。
- ③ 如果有一个操作数为 NaN，结果将为 NaN。
- ④ +0 和-0 进行比较时按相等处理。

8. 上溢出、下溢出和非法计算

上溢出将根据结果的符号以+∞或-∞输出。下溢出将根据结果的符号以+0 或-0 输出。  
如果浮点数转换为整数时出现上溢出，那么结果值可能不正确。  
非法计算将导致出现 NaN。

9. 用编程软件 CX-P 读/写浮点数

当用编程软件 CX-P 将 I/O 内存编辑显示器里的数据格式指定为浮点时，用户输入的标准十进制数将会自动转换为 IEEE754 格式的浮点数，并写到 I/O 内存中。当在显示器上进行监控时，浮点数也会自动转换为标准的十进制数。

10. 浮点数转换、运算的标志位

CP1H 进行浮点运算时，要影响如下标志位。

ER(CF003)：出错标志位。

EQ(CF006)：相等标志位。

OF(CF009)：上溢出标志位。

UF(CF010)：下溢出标志位。

N(CF008)：负标志位。

3.13.2 浮点数与二进制数间的转换指令

表 3.28 是浮点数与二进制数之间的转换指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。

表 3.28 浮点数→有符号二进制数转换指令

名 称	梯形图符号	操作数的含义及范围	指令功能及对标志位的影响
浮点数→16 位有符号二进制数指令 FIX/@ FIX		S：源首通道 S+1 和 S 的内容必须是浮点数且整数部分不超出-32768~32767 范围：CIO、W、H、T/C、A、D、@/*D、#等 R：结果首通道 范围：CIO、W、H、T/C、A448~959、D、@/*D、DR 等	执行条件为 ON 时，将 S+1 和 S 中的 32 位浮点数的整数部分转换成 16 位有符号的二进制数据，并把结果存于 R 中 对标志位的影响： ① S+1, S 中的数据为 NaN 或超出范围时，ER 置为 ON ② 和为 0000 时，EQ 为 ON ③ 结果最高位为 1 时，N 为 ON
浮点数→32 位有符号的二进制数指令 FIXL/@FIXL		S：源首通道 S+1 和 S 的内容必须是浮点数且整数部分不超出-2147483648~2147483647 范围：CIO、W、H、T/C、A、D、@/*D、#等 R：结果首通道 范围：CIO、W、H、T/C、A448~958、D、@/*D 等	当执行条件为 ON 时，将 S+1 和 S 中的 32 位浮点数的整数部分转换成 32 位有符号的二进制数据，并把结果存于 R+1 和 R 中 对标志位的影响同上
16 位有符号的二进制数→浮点数指令 FLT/@ FLT		S：源首通道 S 的内容为-32768~32767(十进制)之间的有符号二进制数据 范围：CIO、W、H、T/C、A、D、@/*D、DR、#等 R：结果首通道 范围：CIO、W、H、T/C、A448~958、D、@/*D 等	执行条件为 ON 时，将 S 中的 16 位有符号的二进制数转换成 32 位浮点数，并把结果存于 R+1 和 R 中。浮点数的小数点后加一个 0 对标志位的影响： ① 转换结果的指数部和尾数部均为 0 时，EQ 为 ON ② 结果变为负数时，N 为 ON
32 位有符号的二进制数→浮点数指令 FLTL/@ FLTL		S：源首通道 S+1 和 S 内容-2147483648~2147483647(十进制)之间的有符号二进制数据 范围：CIO、W、H、T/C、A、D、@/*D、#等 R：结果首通道 范围：CIO、W、H、T/C、A448~958、D、@/*D 等	执行条件为 ON 时，将 S+1 和 S 中的 32 位有符号的二进制数转换成 32 位浮点数，并把结果存于 R+1 和 R 中。浮点数的小数点后加一个 0 对标志位的影响同上

### 1. 浮点数→16 位有符号的二进制数指令 FIX(450)

浮点数→16 位有符号的二进制数 FIX(450) 示例一如图 3.71 所示。H1、H0 中的内容为 40F00000H，执行图 3.71 所示程序后，结果通道 H10 中的内容为 0007H。

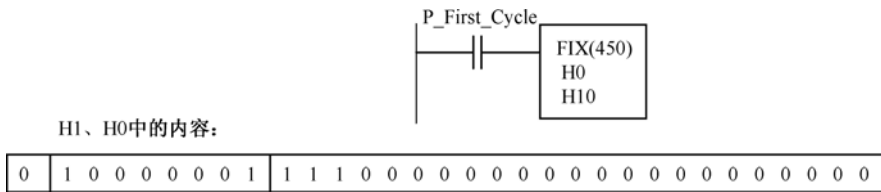


图 3.71 浮点数→16 位有符号二进制数示例一

分析如下：

H1、H0 表示的浮点数如下。

符号 s: 0

指数 e:  $2^7 + 2^0 = 129$

实际指数:  $129 - 127 = 2$

尾数 f:  $0.5 + 0.25 + 0.125 = 0.875$

实际尾数:  $1 + 0.875 = 1.875$

故该浮点数实际值为:  $(-1)^0 \times 2^2 \times 1.875 = 7.5$

其整数部分为 7，执行程序后，将 7 转换为二进制数，所以，H10 中的内容为 0007H。

浮点数→16 位有符号的二进制数 FIX(450) 示例二如图 3.72 所示。设 H1、H0 中的内容为 C0600000H，当执行图 3.72 所示程序后，H10 中的内容为 FFFDH (FFFDH 是 -3 的补码)。

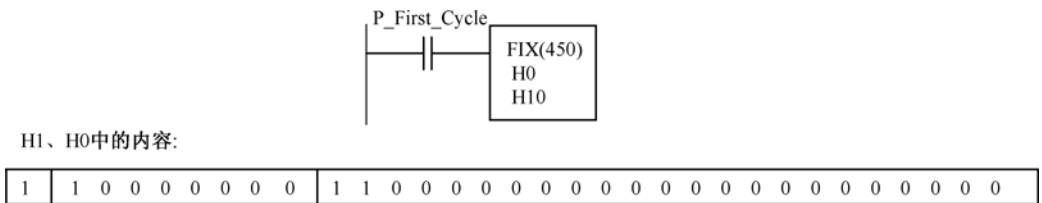


图 3.72 浮点数→16 位有符号二进制数示例二

分析如下：

H1、H0 表示的浮点数如下。

符号 s: 1

指数 e:  $2^7 = 128$

实际指数:  $128 - 127 = 1$

尾数 f:  $0.5 + 0.25 = 0.75$

实际尾数:  $1 + 0.75 = 1.75$

故该浮点数的实际数值为:  $(-1)^1 \times 2^1 \times 1.75 = -3.5$

-3.5 的整数部分是 -3，执行程序后将 -3 转换成二进制数，结果通道 H10 中的内容为 FFFDH (FFFDH 是 -3 的补码)。

可见，执行 FIX 指令时，结果通道中的有符号二进制数是用补码表示的。

2. 16 位有符号的二进制数→浮点数指令 FLT(452)

16 位有符号的二进制数→浮点数指令 FLT(452) 示例如图 3.73 所示。设 H0 中的内容为 FFFAH(-6 的补码)，H0 中的内容是-6 的补码，即被转换数是-6，程序执行后，结果通道 H11、H10 中的内容是-6 的浮点数，其内容为 C0C0 0000H，如图 3.73 所示。

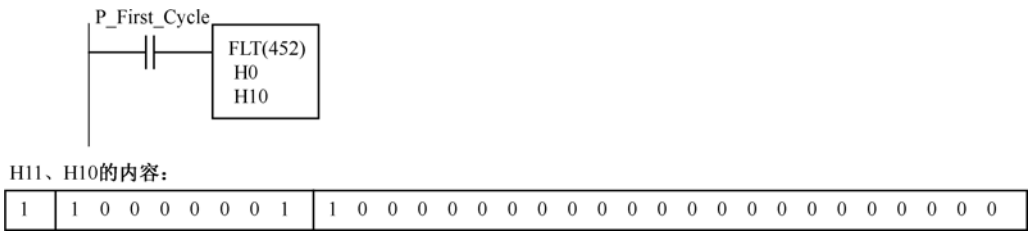


图 3.73 16 位二进制→浮点数示例

对 H11、H10 中的内容进行如下验证。

符号 s: 1  
指数 e:  $2^7+2^0=129$   
实际指数:  $129-127=2$   
尾数 f: 0.5  
实际尾数:  $1+0.5=1.5$   
故该浮点数的实际数值为:  $(-1)^1 \times 2^2 \times 1.5 = -6$   
注意，当源通道中的数据是负数时，要用补码表示。

3.13.3 浮点数加+F、减-F、乘\*F、除/F 运算指令

表 3.29 是浮点数加、减、乘、除运算指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。表 3.29 中指令均具有上升沿微分功能，为简洁只画出梯形图指令的原来形式。

表 3.29 浮点数相加运算指令

名 称	梯形图符号	操作数的含义及范围	指令功能及对标志位的影响
浮点数加法运算指令 +F/@+F		S1: 被加浮点数首通道 S2: 加浮点数首通道 S1、S2 范围: CIO、W、H、T/C、A、D、@/*D、#等 R: 结果首通道 范围: CIO、W、H、T/CA、A448~958、D、@/*D 等	执行条件为 ON 时，将 S1+1、S1 与 S2+1、S2 中的 32 位浮点数相加，结果存于 R+1、R 中 对标志位的影响： ① 被加数或加数为非数时，ER 为 ON ② +∞加-∞时，ER 为 ON ③ 转换结果的指数部和尾数部均为 0 时，EQ 为 ON ④ 结果变为负数时，N 为 ON ⑤ 若结果的绝对值太大而不能表示为 32 位浮点数，OF 为 ON，运算结果以±∞输出 ⑥ 若结果的绝对值太小而不能表示为 32 位浮点数，UF 为 ON，运算结果以浮点数 0 输出
浮点数减法运算指令 -F/@-F		S1: 被减浮点数首通道 S2: 减浮点数首通道 R: 结果首通道 范围同上	执行条件为 ON 时，将 S1+1、S1 与 S2+1、S2 中的 32 位浮点数相减，结果存于 R+1 和 R 中 对标志位的影响： ① 被减数或减数为非数时，ER 为 ON ② +∞和+∞或-∞和-∞进行减法运算时，ER 为 ON ③④⑤⑥同上

(续表)

名 称	梯形图符号	操作数的含义及范围	指令功能及对标志位的影响
浮点数乘法运算指令 *F/@*F	<div><div>*F(456)</div><div>S<sub>1</sub></div><div>S<sub>2</sub></div><div>R</div></div>	S <sub>1</sub> : 被乘浮点数首通道 S <sub>2</sub> : 乘浮点数首通道 R: 结果首通道 范围同上	执行条件为 ON 时, 将 S <sub>1</sub> +1、S <sub>1</sub> 与 S <sub>2</sub> +1、S <sub>2</sub> 中的 32 位浮点数相乘, 结果存于 R+1 和 R 中 对标志位的影响: ① 被乘数或乘数为非数时, ER 为 ON ② 对 0 和 +∞或-∞进行乘法运算时, ER 为 ON ③④⑤⑥同上
浮点数除法运算指令 /F/@/F	<div><div>/F(457)</div><div>S<sub>1</sub></div><div>S<sub>2</sub></div><div>R</div></div>	S <sub>1</sub> : 被除浮点数首通道 S <sub>2</sub> : 除浮点数首通道 R: 结果首通道 范围同上	执行条件为 ON 时, 将 S <sub>1</sub> +1、S <sub>1</sub> 中的 32 位浮点数与 S <sub>2</sub> +1、S <sub>2</sub> 中的 32 位浮点数相除, 结果存于 R+1 和 R 中 对标志位的影响: ① 被除数或除数为非数时, ER 为 ON ② 被除数、除数均为 0、或+/-∞时, ER 为 ON ③④⑤⑥同上

1. 浮点数加运算指令+F(454)

被加数和加数不同组合产生的结果见表 3.30。

表 3.30 被加数和加数不同组合产生的结果

加 数	被 加 数				
	0	数字	+∞	-∞	NaN
0	0	数字	+∞	-∞	见注②
数字	数字	见注①	+∞	-∞	
+∞	+∞	+∞	+∞	见注②	
-∞	-∞	-∞	见注②	-∞	
NaN	见注②				见注②

注：① 结果可能为 0(包括下溢出)、一个数字、+∞或-∞。

② 错误标志 ER 将变为 ON 且指令不执行。

浮点数加运算指令例子如图 3.74 所示。设 H1、H0 中的内容为 C0C0 0000H(-6 的浮点数)，H3、H2 中的内容为 40E0 0000H(7 的浮点数)，即图 3.74(a) 的程序是进行 (-6)+7 的运算，当执行如图 3.74(a) 所示程序后，结果通道中应该是 1 的浮点数，所以执行程序后 H11、H10 中的内容为 3F80 0000H，如图 3.74(b) 所示。

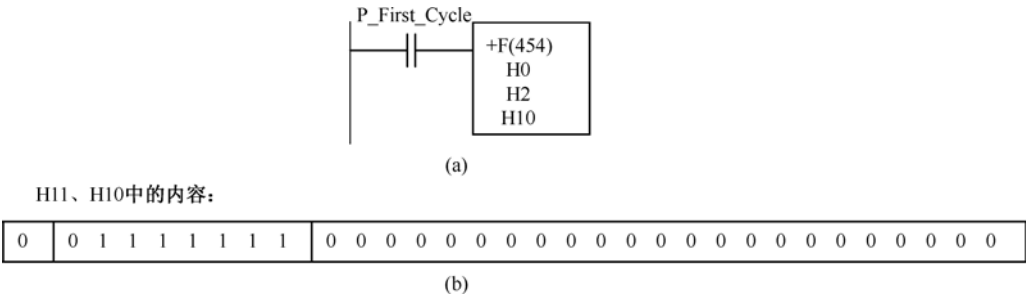


图 3.74 浮点数相加示例

对 H11、H10 中的内容验证如下。

符号 s: 0

指数 e:  $2^6+2^5+2^4+2^3+2^2+2^1+2^0=127$

实际指数:  $127-127=0$

尾数 f: 0  
实际尾数:  $1+0=1.0$   
故该浮点数的实际数值为:  $(-1)^0\times2^0\times1.0=1$

2. 浮点数减运算指令-F(455)

被减数和减数不同组合产生的结果见表 3.31。

表 3.31 被减数和减数不同组合产生的结果

减 数	被 减 数				
	0	数字	$+\infty$	$-\infty$	NaN
0	0	数字	$+\infty$	$-\infty$	见注②
数字	数字	见注①	$+\infty$	$-\infty$	
$+\infty$	$-\infty$	$-\infty$	见注②	$-\infty$	
$-\infty$	$+\infty$	$+\infty$	$+\infty$	见注②	
NaN	见注②				见注②

注：① 结果可能为 0(包括下溢出)、一个数字、 $+\infty$ 或 $-\infty$ 。  
② 错误标志 ER 将变为 ON 且指令不执行。

浮点数减运算指令例子如图 3.75 所示。设 H1、H0 中的内容为 C0C0 0000H(−6 的浮点数)，H3、H2 中的内容为 40E0 0000H(7 的浮点数)，即图 3.75 (a) 的程序是进行−6−7 的运算，当执行如图 3.75 (a) 所示程序后，结果通道中应该是−13 的浮点数，即结果通道 H11、H10 中的内容为 C150 0000H，如图 3.75 (b) 所示。

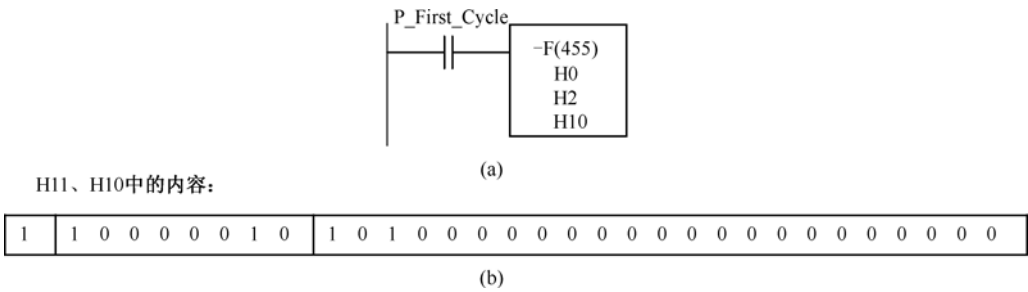


图 3.75 浮点数相减示例

对 H11、H10 中的内容验证如下。  
符号 s: 1  
指数 e:  $2^7+2^1=130$   
实际指数:  $130-127=3$   
尾数 f:  $0.5+0.125=0.625$   
实际尾数:  $1+0.625=1.625$   
故该浮点数的实际数值为:  $(-1)^1\times2^3\times1.625=-13$

3. 浮点数乘运算指令\*F(456)

被乘数和乘数不同组合时产生的结果见表 3.32。



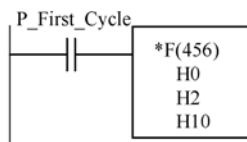
表 3.32 被乘数和乘数不同组合产生的结果

乘 数	被 乘 数				NaN
	0	数字	$+\infty$	$-\infty$	
0	0	0	见注②	见注②	见注②
数字	0	见注①	$+/-\infty$	$+/-\infty$	
$+\infty$	见注②	$+/-\infty$	$+\infty$	$-\infty$	
$-\infty$	见注②	$+/-\infty$	$-\infty$	$+\infty$	
NaN	见注②				见注②

注: ① 结果可能为 0(包括下溢出)、一个数字、 $+\infty$ 或 $-\infty$ 。

② 错误标志 ER 将变为 ON 且指令不执行。

浮点数乘运算指令例子如图3.76所示。设 H1、H0 中的内容为 C0C0 0000H(-6 的浮点数)，H3、H2 中的内容为 40E0 0000H(7 的浮点数)，即图 3.76(a) 的程序是进行  $(-6) \times 7$  的运算，当执行如图 3.76(a) 所示程序后，结果通道中应该是 -42 的浮点数，即结果通道 H11、H10 中的内容为 C228 0000H，如图 3.76(b) 所示。



(a)

H11、H10中的内容:

[illegible]

(b)

图 3.76 浮点数相乘示例

对 H11、H10 中的内容验证如下。

符号 s: 1

指数 e:  $2^7+2^2=132$

实际指数:  $132-127=5$

尾数 f:  $0.25+0.0625=0.3125$

实际尾数:  $1+0.3125=1.3125$

故该浮点数的实际值为： $(-1)^1 \times 2^5 \times 1.3125 = -42$

#### 4. 浮点数除运算指令/F(457)

被除数和除数不同组合时产生的结果见表 3.33。

表 3.33 被除数和除数不同组合产生的结果

除数	被除数				
	0	数字	$+\infty$	$-\infty$	NaN
0	见注③	$+\infty$	$+\infty$	$-\infty$	见注③
数字	0	见注①	$+\infty$	$+\infty$	
$+\infty$	0	见注②	见注③	见注③	
$-\infty$	0	见注②	见注③	见注③	
NaN	见注③				

注：① 结果可能为 0(包括下溢出)、一个数字、 $+\infty$ 或 $-\infty$ 。

② 下溢出，结果将为 0。

③ 错误标志 ER 将变为 ON 且指令不执行。

浮点数除运算指令例子如图 3.77 所示。设 H1、H0 中的内容为 40E0 0000H(7 的浮点数)，H3、H2 中的内容为 40E0 0000H(7 的浮点数)，即图 3.77(a)的程序是进行  $7 \div 7$  的运算，当执行如图 3.77(a)所示程序后,结果通道中应该是 1 的浮点数,即结果通道 H11、H10 中的内容为 3F80 0000H,如图 3.77(b)所示。

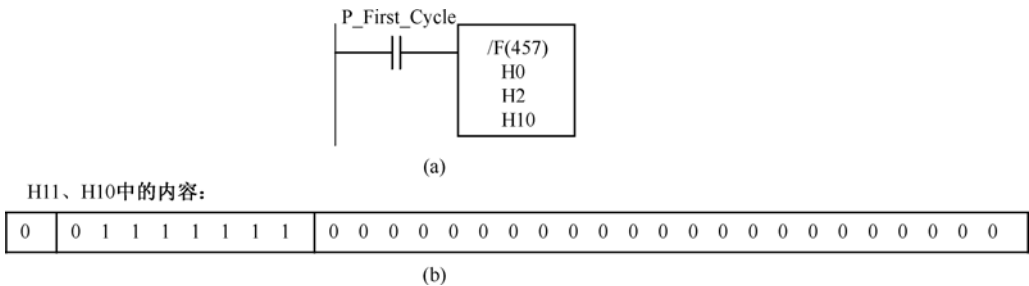


图 3.77 浮点数相除运算

对 H11、H10 中的内容验证如下。

符号 s: 0

指数 e:  $2^6+2^5+2^4+2^3+2^2+2^1+2^0=127$

实际指数:  $127-127=0$

尾数 f: 0.0

实际尾数:  $1+0.0=1.0$

故该浮点数对应的实际值为:  $(-1)^0 \times 2^0 \times 1.0 = 1$

3.13.4 角度↔弧度变换指令

表 3.34 是角度→弧度变换指令操作数的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。

表 3.34 度→弧度变换指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行时对标志位的影响
度→弧度 变换指令 RAD/@ RAD	<div><div>RAD(458) S R</div><div>@RAD(458) S R</div></div>	S: 是源首通道 范围: CIO、W、H、T/C、A、D、 @/*D、#等 R: 结果首通道 范围: CIO、W、H、T/C、A448~958、 D、@/*D 等	执行条件为 ON 时, 将 S+1、S 中的 32 位浮点数角度转换成浮 点数弧度, 并把结果存于 R+1、R 中 对标志位的影响: ① 角度为非数时, ER 为 ON ② 转换结果的指数部和尾数部均为 0 时, EQ 为 ON ③ 结果变为负数时, N 为 ON ④ 若结果的绝对值太大而不能表示为 32 位浮点数, OF 为 ON ⑤ 若结果的绝对值太小而不能表示为 32 位浮点数, UF 为 ON
弧度→度 变换指令 DEG/@ DEG	<div><div>DEG(459) S R</div><div>@DEG(459) S R</div></div>	同上	执行条件为 ON 时, 将 S+1、S 中的 32 位浮点数弧度转换成浮 点数角度, 并把结果存于 R+1、R 中 对标志位的影响同上

1. 度→弧度变换指令 RAD(458)

度→弧度指令例子如图3.78所示。设 H1、H0 中的内容为 4334 0000H(180° 的浮点数)，180° 对应  $\pi$  弧度，当执行如图3.78(a)所示程序后，结果通道 H11、H10 中应是  $\pi$  弧度的浮点数，内容为 4049 0FDBH，如图3.78(b)所示。

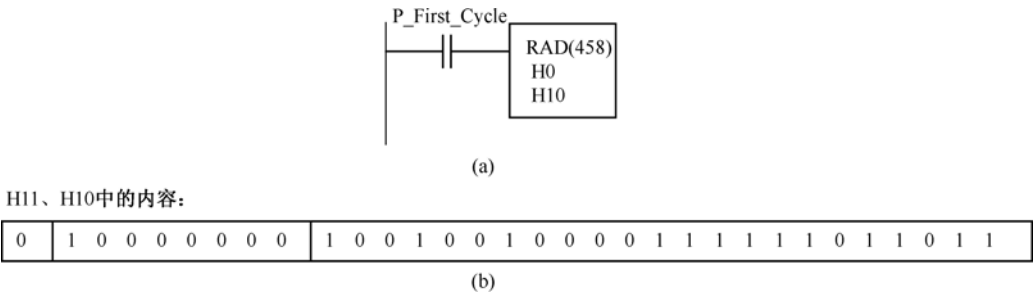


图 3.78 度→弧度示例

对 H11、H10 中的内容验证如下。

符号 s: 0

指数 e:  $2^7=128$

实际指数:  $128-127=1$

尾数 f:  $(2^{22}+2^{19}+2^{16}+2^{11}+2^{10}+2^9+2^8+2^7+2^6+2^4+2^3+2^1+2^0) \times 2^{-23} \approx 0.5707964$

实际尾数:  $1+0.5707964=1.5707964$

故该浮点数的实际数值为:  $(-1)^0 \times 2^1 \times 1.5707964 \approx 3.141593$

2. 弧度→度变换指令 DEG(459)

弧度→度指令例子如图3.79所示。设 H1、H0 中的内容为 4049 0FDBH( $\pi$  弧度的浮点数)， $\pi$  弧度对应 180°，当执行如图3.79(a)所示程序后，结果通道 H11、H10 中应是 180° 的浮点数，内容为 4334 0000H，如图3.79(b)所示。

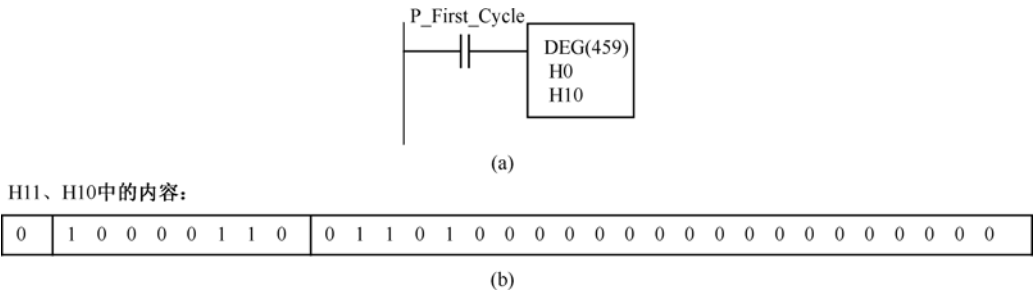


图 3.79 弧度→度示例

对 H11、H10 中的内容验证如下。

符号 s: 0

指数 e:  $2^7+2^2+2^1=128+4+2=134$

实际指数:  $134-127=7$

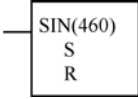
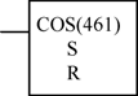
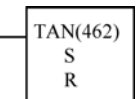
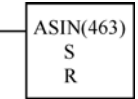
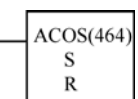
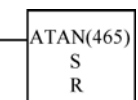
尾数 f:  $0.25+0.125+0.03125=0.40625$

实际尾数：1+0.40625=1.40625  
故该浮点数的实际数值为： $(-1)^0 \times 2^7 \times 1.40625=180$

3.13.5 三角函数运算指令

表 3.35 是三角函数运算指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响，所有指令均具有微分功能@，微分形式梯形图省略不再列出。

表 3.35 三角函数运算指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行时对标志位的影响
正弦运算指令 SIN/@ SIN		S: 源数据首通道 源数据不超出-65535~65535	执行条件为 ON 时，计算 S+1、S 中 32 位浮点数弧度的正弦，结果存于 R+1、R 中 对标志位的影响： ① 源数据为非数或超出范围时，ER 为 ON ② 结果指数部和尾数部均为 0 时，EQ 为 ON ③ 运算结果变为负数时，N 为 ON
余弦运算指令 COS(461)/@COS(461)		范围：CIO、W、H、T/C、A、D、@/*D、#等 R: 结果数据首通道	执行条件为 ON 时，计算 S+1、S 中 32 位浮点数弧度的余弦，结果存于 R+1、R 中 对标志位的影响同上
正切运算指令 TAN(462)/@TAN(462)		范围：CIO、W、H、T/C、A448~958、D、@/*D 等	执行条件为 ON 时，计算 S+1、S 中 32 位浮点数弧度的正切，结果存于 R+1、R 中 对标志位的影响： ①②③同上 ④ 若结果的绝对值太大而不能表示为 32 位浮点数，OF 为 ON
反正弦运算指令 ASIN/@ ASIN		S: 源数据首通道 源数据不超出-1.0~+1.0 范围：CIO、W、H、T/C、A、D、@/*D、#等 R: 结果数据首通道	执行条件为 ON 时，计算 S+1、S 中 32 位浮点数表示的正弦值对应的弧度，结果 $(-\pi/2 \sim \pi/2)$ 存于 R+1、R 中 对标志位的影响： ① 源数据为非数或超出范围时，ER 为 ON ② 运算结果指数部和尾数部均为 0 时，EQ 为 ON ③ 运算结果变为负数时，N 为 ON
反余弦运算指令 ACOS/@ ACOS		范围：CIO、W、H、T/C、A448~958、D、@/*D 等	执行条件为 ON 时，计算 S+1、S 中 32 位浮点数表示的余弦值对应的弧度，结果 $(-\pi/2 \sim \pi/2)$ 存于 R+1、R 中 对标志位的影响： ① 源数据为非数或超出范围时，ER 为 ON ② 运算结果指数部和尾数部均为 0 时，EQ 为 ON
反正切运算指令 ATAN/@ ATAN		S: 源数据首通道 范围：CIO、W、H、T/C、A、D、@/*D、#等 R: 结果数据首通道 范围：CIO、W、H、T/C、A448~958、D、@/*D	执行条件为 ON 时，计算 S+1、S 中 32 位浮点数表示的正切值对应的弧度，结果 $(-\pi/2 \sim \pi/2)$ 存于 R+1、R 中 对标志位的影响： ① 源数据为非数时，ER 为 ON ② 运算结果指数部和尾数部均为 0 时，EQ 为 ON ③ 运算结果变为负数时，N 为 ON

1. 正弦运算指令 SIN(460)

正弦运算指令示例如图3.80所示。设 H1、H0 中的内容为 4049 0FDBH( $\pi$  弧度的浮点数)， $\pi$  弧度对应正弦值为 0，当执行如图3.80(a)所示程序后，结果通道 H11、H10 中应是 0 的浮点数，内容为 8000 0000H，如图3.80(b)所示。  
对 H11、H10 中的内容验证如下。



3. 正切运算指令 TAN(462)

正切运算指令例子如图3.82所示。设 H1、H0 中的内容为 4234 0000H(45° 浮点数)，当执行如图 3.82(a) 所示程序后，RAD 把 45° 转换成 $\pi/4$ ，TAN 的结果通道 H21、H20 中的数据是 $\pi/4$  正切值(1) 的浮点数，内容为 3F80 0000H，如图3.82(b)所示。

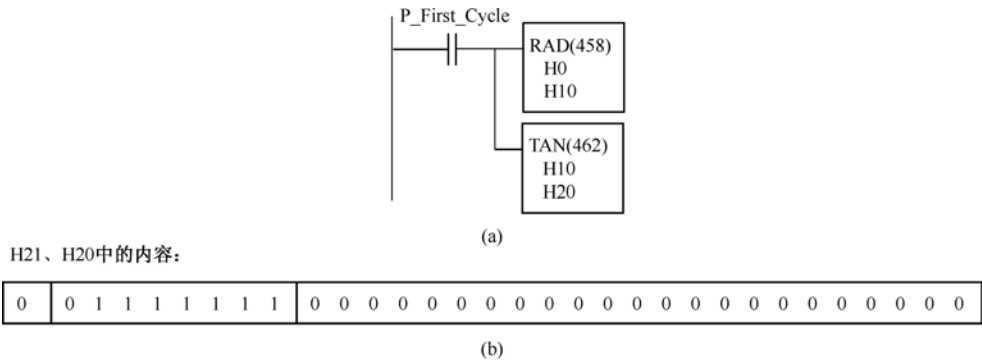


图 3.82 正切运算示例

对 H21、H20 中的内容验证如下。

符号 s: 0  
指数 e: 127  
实际指数: 127-127=0  
尾数 f: 0  
实际尾数: 1+0=1  
故该浮点数的实际数值为:  $(-1)^0 \times 2^0 \times 1 = 1$

4. 反正弦运算指令 ASIN(463)

反正弦运算指令例子如图3.83所示。设 H1、H0 中的内容为 3F00 0000H(0.5 的浮点数)，当执行如图 3.83(a) 所示程序后，结果通道 H11、H10 中是 0.5 的反正弦值( $\pi/6$ ) 的浮点数，其内容为 3F060A92H，如图3.83(b)所示。

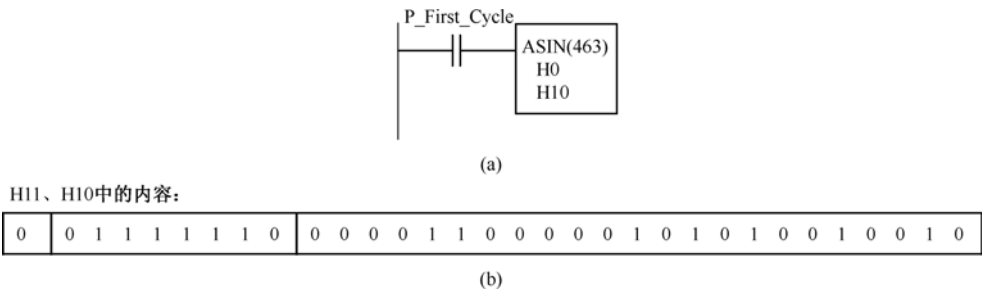


图 3.83 反正弦运算示例

对 H11、H10 中的内容验证如下。

符号 s: 0  
指数 e: 126  
实际指数: 126-127=-1

尾数 f:  $(2^{18}+2^{17}+2^{11}+2^9+2^7+2^4+2^1) \times 2^{-23} \approx 0.0471976$   
实际尾数:  $1+0.0471976=1.0471976$   
故该浮点数的实际数值为:  $(-1)^0 \times 2^{-1} \times 1.0471976= 0.5235988 (30^\circ \text{ 的弧度数})$

3.13.6 平方根 SQRT、指数 EXP、对数 LOG 指令

表3.36是平方根、指数、对数运算指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。

表 3.36 平方根、指数、对数运算指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行时对标志位的影响
平方根运算指令 SQRT/@SQRT	<div><div>SQRT(466) S R</div><div>@SQRT(466) S R</div></div>	S: 源数据首通道 范围: CIO、W、H、T/C、A、 D、@/*D、#等 R: 结果首通道 范围: CIO、W、H、T/C、 A448~958、D、@/*D	执行条件为 ON 时, 计算 S+1、S 中 32 位浮点数的平方根, 结果存于 R+1、R 中 对标志位的影响: ① 源数据为负数或非数时, ER 为 ON ② 结果指数部和尾数部均为 0 时, EQ 为 ON ③ 结果的绝对值太大而不能表示为 32 位浮点数, OF 为 ON, 运算结果以+∞输出
指数运算指令 EXP/@ EXP	<div><div>EXP(467) S R</div><div>@ EXP(467) S R</div></div>		执行条件为 ON 时, 计算 S+1、S 中 32 位浮点数的自然指数, 即计算 $e^x$ (X 是源数据) 的值, 结果存于 R+1、R 中 对标志位的影响: ① 源数据为非数时, ER 为 ON ② 结果指数部和尾数部均为 0 时, EQ 为 ON ③ 结果的绝对值太大而不能表示为 32 位浮点数, OF 为 ON, 运算结果以+∞输出 ④ 若结果的绝对值太小而不能表示为 32 位浮点数, UF 为 ON, 运算结果以 0 输出
对数运算指令 LOG/@LOG	<div><div>LOG(468) S R</div><div>@LOG(468) S R</div></div>		执行条件为 ON 时, 计算 S+1、S 中的 32 位浮点数的自然对数, 即计算 $\ln X$ (X 是源数据) 的值, 结果存于 R+1、R 中 对标志位的影响: ① 源数据为负数或非数时, ER 为 ON ② 结果指数部和尾数部均为 0 时, EQ 为 ON ③ 结果的绝对值太大而不能表示为 32 位浮点数, OF 为 ON, 运算结果以±∞输出 ④ 运算结果变为负数时, N 为 ON

1. 平方根运算指令 SQRT (466)

平方根运算指令示例如图3.84 所示。设 H1、H0 中的内容为 42C8 0000H(100 的浮点数), 当执行如图3.84 (a)所示程序后, 结果通道 H11、H10 中是 100 的平方根(10)的浮点数, 其内容为 41200000H, 如图3.84 (b)所示。

对 H11、H10 中的内容验证如下。  
符号 s: 0  
指数 e:  $2^7+2^1=130$   
实际指数:  $130-127=3$   
尾数 f: 0.25  
实际尾数:  $1+0.25=1.25$   
故该浮点数的实际数值为:  $(-1)^0 \times 2^3 \times 1.25=10$





对 H11、H10 中的内容验证如下。

符号 s: 0

指数 e: 0

尾数 f: 0

故该浮点数的实际数值为 0。

### 3.13.7 浮点数运算的应用举例

下面的例子中使用了多种浮点运算指令。阅读此例程序时, 请注意各种浮点数运算指令的功能和使用方法。

用 PLC 控制刻刀在部件上的运动轨迹, 可以刻出不同的图形。刻刀某时刻在平面上的位置如图 3.87 所示。图 3.87 中, 黑点为刻刀所在位置, 黑点与规定原点的距离是  $\gamma$ , 其在横轴和纵轴上的投影分别是  $X$  和  $Y$ ,  $\gamma$  与横轴的夹角是  $\alpha$ 。

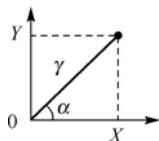


图 3.87 被测点的平面坐标

当确定了  $X$  和  $Y$  的数值后, 可以计算出  $\gamma$  和  $\alpha$ , 图 3.88 是计算程序。  $X$  和  $Y$  是 4 位 BCD 数,  $\gamma$  和  $\alpha$  是 BCD 数。  $X$ 、 $Y$  的 BCD 数据分别放在 D0、D1 中。  $\gamma$ 、 $\alpha$  的 BCD 数据分别放在 D100、D101 中, 结果中, 小数点后面的部分被截掉。  $\gamma$ 、 $\alpha$  的运算公式及存储地址如下:

$$\gamma = \sqrt{X^2 + Y^2}, \quad \gamma = \sqrt{100^2 + 100^2} = 141.4214$$

$$\alpha = \arctan(Y/X), \quad \alpha = \arctan(100/100) = 45^\circ$$

DM 区内容

D0000	<div style="border: 1px solid black; padding: 2px; display: inline-block;">0100</div>	$X$		D0100	<div style="border: 1px solid black; padding: 2px; display: inline-block;">0141</div>	$\gamma$
	(BCD)		→		(BCD)	
D0001	<div style="border: 1px solid black; padding: 2px; display: inline-block;">0100</div>	$Y$		D0101	<div style="border: 1px solid black; padding: 2px; display: inline-block;">0045</div>	$\alpha$
	(BCD)				(BCD)	

程序中采用了各种浮点运算指令, 有多种数据转换操作。图 3.88 的程序说明如下。

- ① 将  $X$  和  $Y$  的 BCD 数据转换成 16 位二进制数。
- ② 将  $X$  和  $Y$  的 16 位二进制数转换成浮点数。
- ③ 计算  $\gamma$  的浮点数值:
  - 用指令 \*F 分别计算  $X^2$  和  $Y^2$  的浮点数值。
  - 用指令 +F 计算  $X^2 + Y^2$  的浮点数值。
  - 用指令 SQRT 计算  $\sqrt{X^2 + Y^2}$  的值。
- ④ 计算  $\alpha$ 。
  - 用指令 /F 计算  $Y/X$  的浮点数值。
  - 用指令 ATAN 计算  $\arctan Y/X$ , 得到  $\alpha$  的弧度浮点数。
  - 用指令 DEG 将  $\alpha$  的弧度浮点数转换成度的浮点数。
- ⑤ 用指令 FIX 将  $\gamma$  和  $\alpha$  的浮点数中的整数部分转换成 16 位二进制数。
- ⑥ 将  $\gamma$  和  $\alpha$  的 16 位二进制数转换成 BCD 数值。

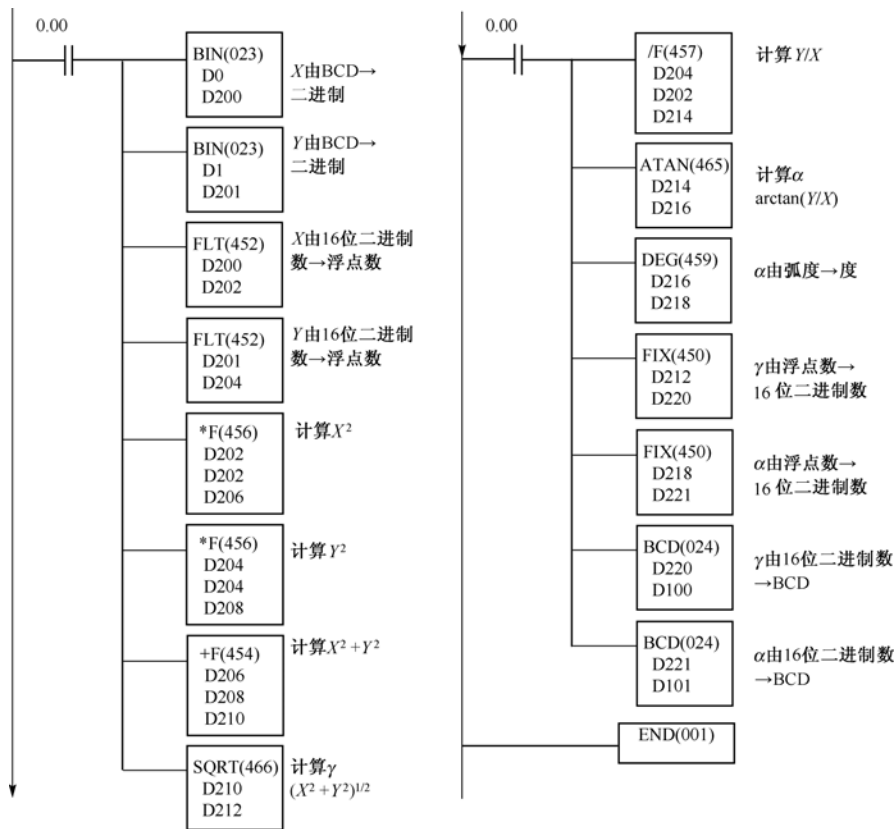


图 3.88 浮点运算指令应用举例

3.14 子程序控制指令

在编写应用程序时，有的程序段要多次重复使用。这样的程序段可以编成一个子程序，在满足一定条件时，中断主程序而转去执行子程序，子程序执行完毕，再返回断点处继续执行主程序。另外，有的程序段不仅要多次使用，而且要求程序段的结构不变、但每次输入和输出的操作数不同。对这样的程序段也可以编成一个子程序，在满足执行条件时，中断主程序的执行而转去执行子程序，并且每次调用时赋予该子程序不同的输入和输出操作数，子程序执行完毕再返回断点处继续执行主程序。

调用子程序与前面介绍的跳转指令都能改变程序的流向，利用这类指令可以实现某些特殊的控制，并具有简化编程、减少程序扫描时间的作用。

表 3.37 是子程序控制指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。

表 3.37 子程序控制指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位影响
子程序调用指令 SBS/@SBS	<div><div>SBS(091)</div><div>N</div><div>@SBS(091)</div><div>N</div></div>	N: 子程序编号，取值为十进制常数 000~255	执行条件为 ON 时,调用并执行 N 编号子程序(SBN~RET 间区域)，之后返回本指令的下一条指令 在下列情况之一时，ER 为 ON： ① 被调用的子程序不存在 ② 子程序自调用 ③ 调用执行中的子程序 ④ 被调用的子程序与调用指令不在同一任务之内 ⑤ 子程序嵌套超过 16 级

(续表)

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位影响
子程序开始指令 子程序返回指令 SBN~RET	<div><div>SBN(092) N</div><div>RET(093)</div></div>	同上	执行子程序调用 SBS N 指令时，调用同一个 N 号的 SBN N 和 RET 指令之间的部分，两条指令要成对使用。SBN 指令定义子程序的开始，RET 表示子程序结束，RET 指令不带操作数 执行该指令不影响标志位
宏指令 MCRO/ @MCRO	<div><div>MCRO(099) N I1 O1</div><div>@MCRO(099) N I1 O1</div></div>	N: 子程序编号，取值为十进制常数 000~255 I1: 输入数据首通道 范围: CIO、W、H、T/C、A0~444、A448~956、D、@/*D 等 O1: 输出数据首通道 范围: CIO、W、H、T/C、D、A448~956、@/*D 等	用一个子程序 N 代替数个具有相同结构、但操作数不同的子程序。执行条件为 ON 时，停止执行主程序，将输入数据 I1~I1+3 的内容复制到 A600~A603 中，然后调用子程序 N。子程序执行完毕，再将 A604~A607 中的内容传送到 O1~O1+3 中，并返回到 MCRO 指令的下一条指令 对标志位的影响同 SBS 指令

3.14.1 子程序调用 SBS、子程序开始 SBN、子程序返回 RET 指令

SBS 是子程序调用指令，SBN 和 RET 是子程序开始和子程序返回指令。所编写的子程序应该在指令 SBN 和 RET 之间。主程序中，在需要调用子程序的地方安排 SBS 指令。若使用非微分指令 SBS 时，在它的执行条件满足时，每个扫描周期都调用一次子程序。若使用 @SBS 时，只在执行条件由 OFF 变 ON 时调用一次子程序。

所有子程序必须放在主程序之后、END 之前。若子程序之后安排了主程序，则该段主程序不被执行。因为 CPU 扫描用户程序时，只要见到 SBN 则认为主程序结束，在编写程序时一定要注意这一点。图3.89 是子程序调用程序的结构及两次调用子程序的执行过程示意图。这段程序的执行过程是：

在执行主程序段 1 时，若 SBS (091)000 的执行条件为 ON 时，立即停止执行主程序，而转去执行 SBN (092)000 与 RET (093) 之间的 000 号子程序。该子程序执行完毕，返回到调用子程序指令 SBS (091)000 的下一条指令，继续执行主程序段 2。

在执行主程序段 2 时，若 SBS (091)001 的执行条件为 ON 时，立即停止执行主程序，而转去执行 SBN (092)001 与 RET (093) 之间的 001 号子程序。该子程序执行完毕，返回到调用子程序指令 SBS (091)001 的下一条指令，继续执行主程序段 3。

图3.89 使用了非微分指令 SBS，只要 SBS 的执行条件为 ON，每个扫描周期都调用一次子程序。图3.90 是调用子程序的示例一。图中，1.00 是调用子程序指令的执行条件。主程序的内容包括传送数据、用 KEEP 指令产生秒脉冲、调用子程序 005。

当 1.00 为 OFF 时，执行 MOV 指令将# 0001 传送到 W20 通道，使 W20.00 为 ON，其余各位均为 OFF。继续执行 KEEP 指令，由 101.00 输出秒脉冲。若 1.00 为 ON 时，立即转去执行 005 号子程序。

005 号子程序有两个内容：其一，移位寄存器指令的数据输入端是 P\_Off，所以 W20.00 的 ON 状态每隔 1 s 向高位移 1 位。若 1.00 一直为 ON，每个扫描周期都调用子程序，移位将持续进行。当 W20.07 变为 ON、且下一个移位脉冲到来时，W20.00 又成为 ON 并重复上述的移位过程；其二，执行 MOV 指令把 W20 通道的内容传送到 100 通道。

在子程序执行过程中，若 1.00 为 OFF 时，立即停止子程序的执行。例如，当 W20.05 为 ON 时，令 1.00 为 OFF 时，100.05 仍保持 ON 状态，但不移位(子程序不再执行)。这时主程序中的 MOV 指令又将#0001 传送到 W20 通道。当 1.00 再次为 ON、又调用子程序 005 时，100.05 立即 OFF。再执行 SFT 指令时，仍是将 W20.00 的 ON 状态依次向高位移位。

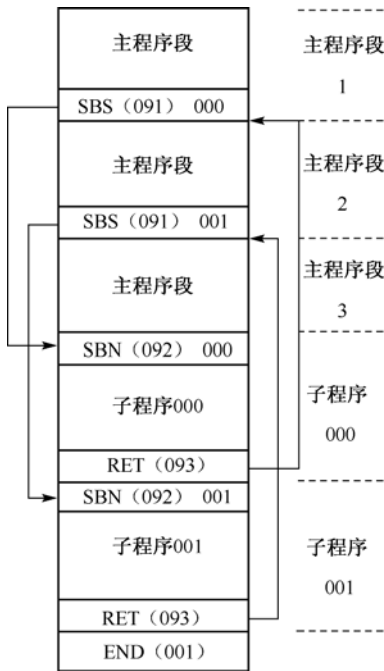


图 3.89 调用子程序的过程

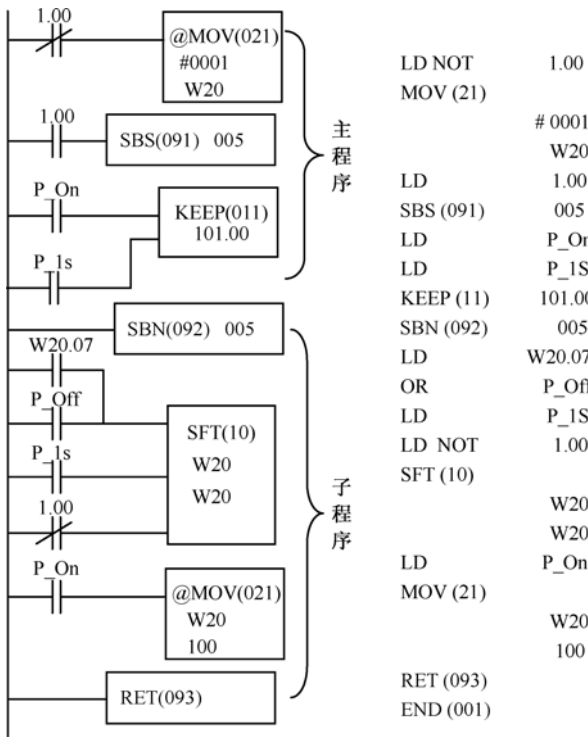


图 3.90 子程序调用示例一

图3.91 是子程序调用的示例二。图中，PLC 加电后经过 4 s, CNT0000 ON 一个扫描周期, 使 100.00 ON (ON 2 s) 并第一次调用子程序。

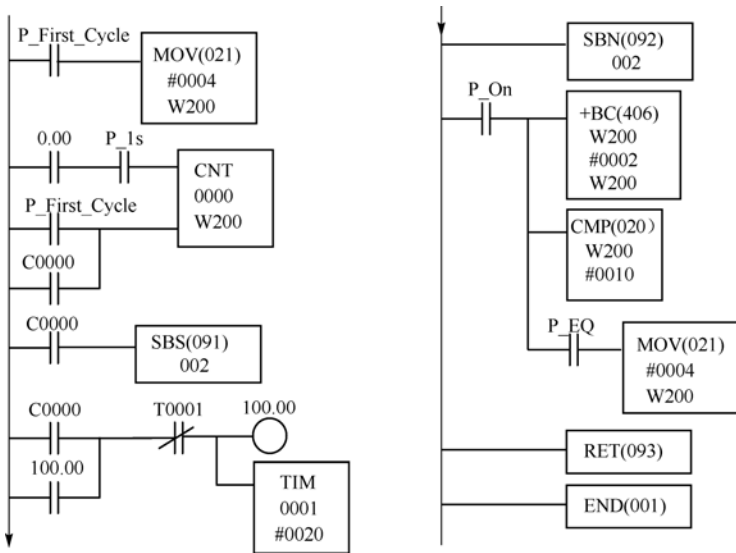


图 3.91 子程序调用示例二

子程序 002 的内容为：其一，将 W200 通道的内容加#0002；其二，将 W200 通道的内容与#0010 比较，若等于#0010 时，向 W200 通道传送#0004。由此可知，每当计数完成标志 C0000 ON 时，其设定值就加#0002。因此，100.00 的 ON 时间总是 2s，而 OFF 时间依次增加 2s。当第 4 次调用子程序时，CNT0000 的设定值又变为#0004，且重复前面程序的执行过程。

图3.92是子程序嵌套调用的例子。图中，D0 中已写入 0100。每当计数完成标志 C0000 ON 时调用子程序 010。执行子程序 010，将 D0 的内容减 1 并与 #0000 比较，当 D0 的内容是 0000 时，再调用子程序 011。执行子程序 011，使 W210.01 ON，返回主程序使 100.01 ON，并使 CNT0000 复位。请读者自行分析该程序的功能。

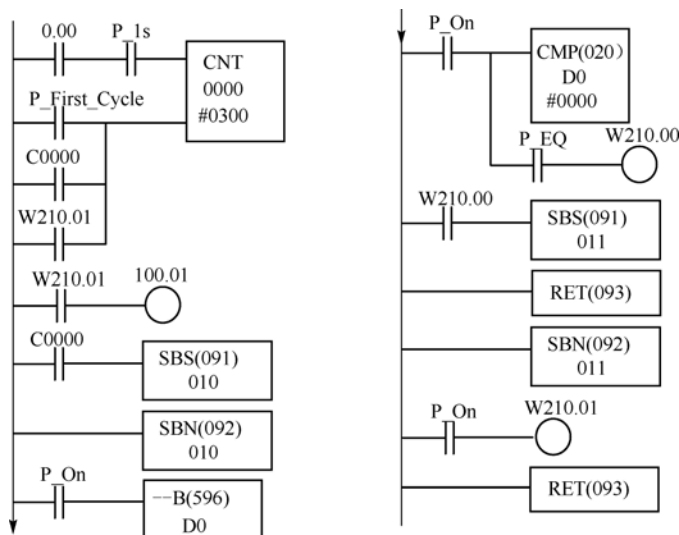


图 3.92 子程序调用示例三

实现此例功能的编程并不是仅此一种，这里只是为了说明子程序嵌套的程序结构形式。

### 3.14.2 宏指令 MCRO

宏指令也是调用子程序的指令，但与前述子程序调用指令有所不同。宏指令的子程序的操作数只是形式上的操作数，在调用子程序时才赋予它们确定的数据。宏指令 MCRO 的操作数 I1 是子程序中第一个输入字的参数，操作数 O1 是子程序中第一个输出字参数，每次调用时，I1 和 O1 的数据可以不同。由于宏指令调用的子程序其输入/输出的数据可以变换，所以提高了子程序存在的价值。

宏指令调用的子程序也是用 SBN/RET 来定义的。与上述子程序的安排相同，子程序必须放在主程序之后、END 指令之前。图3.93是宏指令程序段的安排和宏指令调用的过程示意图。

在使用 MCRO 指令时，通道 A600~A607 已经被系统占用，用户不要再使用这几个通道。

图3.94是使用宏指令的例子。图3.94(a)的梯形图中有两次宏调用，被调用的子程序号是 040。执行两次宏调用与执行图 3.94(b) 程序的功能完全相同。

对图3.94，当 1.00 由 OFF 变为 ON 时执行一次宏调用，第一个输入字是 W200，第一个输出字是 H0。当 1.01 由 OFF 变为 ON 时又执行一次宏调用，第一个输入字 W210，第一个输出字是 H10。总之，每次宏调用，子程序的结构不变，只是输入/输出的参数在变化。

综上所述，编写子程序调用程序时要注意以下几点：

- ① 所有子程序都必须放在主程序之后、END 指令之前。
- ② 主程序调用各子程序的次数没有限制。
- ③ 子程序可以嵌套调用，即子程序中又调用别的子程序，但是嵌套不能超过 16 级，且子程序不能自调用。
- ④ 要特别注意子程序执行完毕的返回地址。



图 3.93 宏指令调用的过程

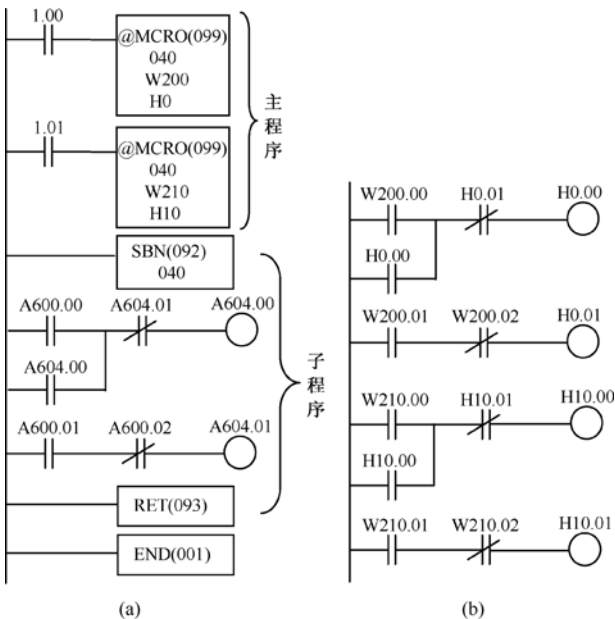


图 3.94 使用宏指令的例子

3.15 步进控制指令

在实际控制中有这样一类情况，其整个控制过程可以分成若干个子过程。当一个子过程结束时立即启动下一个子过程。各子过程的执行像接力一样按一定的顺序进行。对这类控制编写程序时，可以把较大的程序分成若干个程序段，一个程序段称为一个步，每步对应一个实际的子过程，用指令来控制各步执行的顺序，这类指令称为步进控制指令。

CP1H 系列的步进指令 SNXT 和 STEP 就是用于对各步设置断点并执行步进程序的。当执行步进程序时，在执行完上一步、启动下一步之前，将上一步使用的定时器、数据区复位。这样在各步程序中还可以重复使用 PLC 的部分资源。

表 3.38 是步进控制指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。

表 3.38 步进控制指令

名 称	梯形图符号	操作数的含义范围	指令的功能及执行指令对标志位的影响
步启动指令 步定义指令	<div>— <span style="border: 1px solid black; padding: 2px;">SNXT(009) N</span></div> <div>— <span style="border: 1px solid black; padding: 2px;">STEP(008) N</span></div>	B 是步的控制位号，范围： W0.00~W511.15 等	当 SNXT 指令的执行条件为 ON 时，结束上一步的执行、复位上一步用过的定时器和数据区，并启动以 N 为控制位的、以 STEP N 定义的下一个步 对标志位的影响：操作数不在 W 区或 SNXT 用于中断程序时，ER 为 ON
步结束指令	<div>— <span style="border: 1px solid black; padding: 2px;">STEP(008)</span></div>	无操作数	无执行条件 当所有步都执行完毕时，要安排 SNXT N(N 是虚控制位)和 STEP 指令以结束步程序

3.15.1 步启动 SNXT 指令、步定义 STEP 指令

1. 步启动 SNXT(009) 指令

在 STEP 指令之前配置，通过对指定的工序(步)编号进行 OFF→ON，来运行该工序(步)程序(使该工序步进)。CPU 执行到每个步程序段开头的“SNXT(009) N”指令，当在 SNXT 指令之前已存在

某一步程序时，它将结束前一步(N-1)的执行，转而置下一步程序的控制位为 1，执行下一步的程序。并复位前一步使用的定时器、数据区。此时前一步使用的定时器、数据区的状态为：

- ① 用 OUT、OUT NOT 指令指定的继电器为 OFF；
- ② 定时器复位；
- ③ 其他保持。

SNXT(009)指令可以放在步进程序开始、中间和结束。

## 2. 步定义 STEP(008)指令

步定义 STEP(008)指令紧随步启动 SNXT(009)指令之后。

它有两层含义：

① 带操作数的 STEP(008)指令：位于整个步进程序的开始或中间，在 SNXT 指令之后，各工序之前配置，表示指定工序编号的工序开始。它是用来定义紧随此指令之后的步程序开始，它的操作数是该步进程序的控制位。

② 不带操作数的 STEP(008)指令：在最后一个 SNXT 指令之后，整体步程序的最后配置，表示这一系列步进程序块终止。

每个步程序段必须由“SNXT(009) N”指令开头，且紧随一条“STEP(008) N”指令，其中控制位 N 相同。这两条指令后面是该步程序段。步程序段的编写与一般编程一样。

在整个步程序段都编辑完毕，须加一条“SNXT(009) N”指令，该 N 位无特定意义，可用任何未被系统使用的过的工作位号，并在其之后紧跟一条不带控制位的“STEP(008)”指令，表示系列步程序段的结束。

注意：

- ① 一个周期中不能有两个或两个以上的步程序段同时执行。
- ② SNXT 指令和 STEP 指令不能置于子程序、中断程序或块程序段中。
- ③ 不能在步程序段内使用的指令如下。

时序控制指令有 END、IL 和 ILC、JMP 和 JME、JMP0 和 JME0、CJP 和 CJPN。

子程序定义和返回指令有 SBN 和 RET。

## 3. 单步启动标志 A200.12

当 STEP(008)指令定义某一步程序段开始时，单步启动标志 A200.12 ON 1 个扫描周期。A200.12 常被用于复位定时器、计数器等。

### 3.15.2 步进程序的结构及程序的编写规则

#### 1. 步进程序的基本结构

步进程序的基本结构如图 3.95 所示。步进控制程序是由多个步组成的，每一步都是由有执行条件的指令 SNXT(009) N 开始，其后是无执行条件的、用来定义步开始的指令 STEP(008) N，两者的 N 相同。STEP(008) N 指令之后是步的内容。各步编写完毕，要安排一个有执行条件的 SNXT(009) N 指令，指令中的 N 无任何意义，它可以是程序中没有使用过的某一个位号。紧随其后再写一条无执行条件、无操作数的 STEP(008)指令，用以表示全部步的结束。在无操作数的 STEP(008)指令之后还可以安排普通程序。

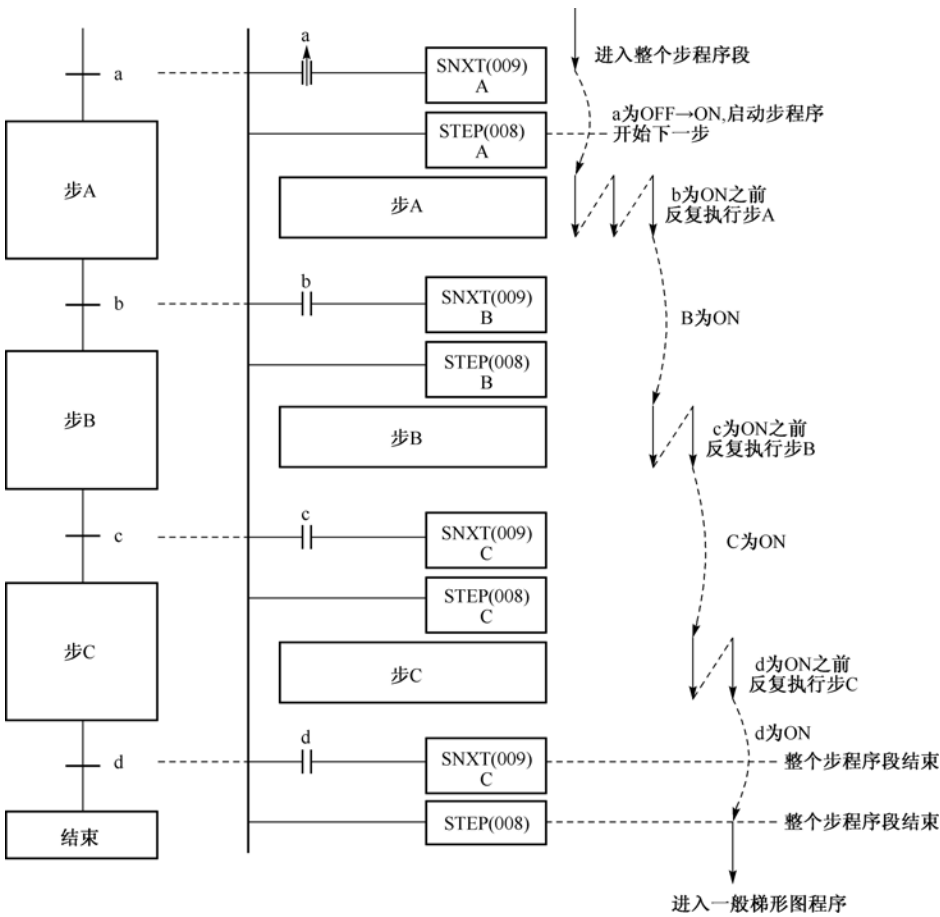


图 3.95 步进程序结构

2. 步进程序结构的几种类型

步进程序的结构可以分为三种类型，即顺序执行类、选择分支执行类和并行分支执行类。图 3.96 是几种类型的步进程序的结构示意图，也称步进程序的流程图。

图3.96(a)是顺序执行类的步进程序结构示意图。这种结构的程序中无分支，前一步结束被清除、复位，后一步即被启动并开始执行，如此一步一步地按顺序执行。图3.95就属于这一类的结构。

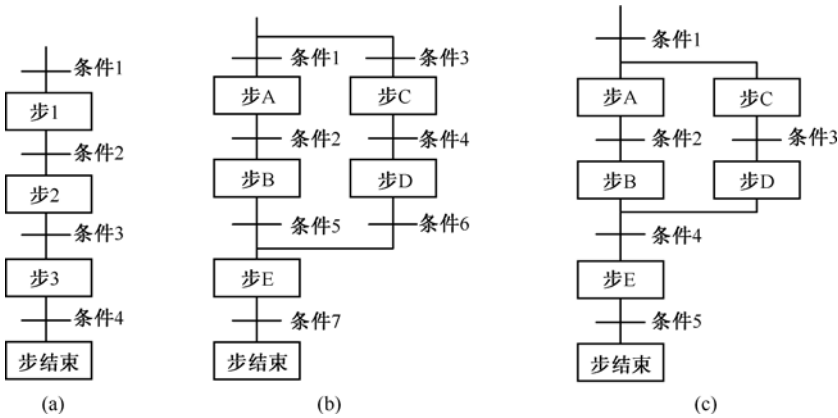


图 3.96 几种类型步进程序结构示意图



图 3.96(b) 是选择分支执行类的步进程序结构示意图。这种结构的程序有几个分支, 每个分支可能有若干个步。在同一时刻只能执行其中的一个分支, 因此几个分支中首步的启动条件一定是互锁的。至于执行哪个分支, 要看哪个分支满足了执行条件。每个分支执行完毕都要去执行同一个步, 如图中步 E。

图 3.96(c) 是并行分支执行类的步进程序结构。这种结构的程序有几个分支, 每个分支可能有若干个步。与选择分支类不同的是, 在满足某个条件时几个分支将同时被启动, 例如图中条件 1 满足时, 步 A、步 C 同时启动。当几个分支都执行完毕时, 又被同一个执行条件所清除, 同时进入下一步。如图中的步 B、步 D 执行完毕, 被条件 4 复位并同时进入步 E。

### 3.15.3 步进程序的执行过程

#### 1. 顺序执行类步进程序的执行过程

以图 3.97 为例, 说明顺序执行类步进程序的执行过程。图中, 当执行条件 0.00 由 OFF→ON→OFF 时, 执行指令 SNXT(009) H0.00, 则由 H0.00 为控制位的步 1 被启动并执行; 当 0.01 由 OFF→ON→OFF 时, 执行指令 SNXT(009) H0.01, 停止执行步 1、复位步 1 所使用的定时器及数据区, 并启动由 H0.01 控制的步 2; 当 0.02 由 OFF→ON→OFF 时, 执行指令 SNXT(009) 0.05 (0.05 无意义) 和 STEP(008), 停止执行步 2 并复位步 2 所使用的定时器及数据区, 步程序全部结束。

#### 2. 选择分支执行类步进程序的执行过程

以图 3.98 为例, 说明选择分支执行类的步进程序的执行过程。图 3.98(a) 是图 3.98(b) 的步程序流程图。从流程图可以看出这是有两个分支的步进程序, 每个分支各有两个步。无论哪个分支, 最后都要执行步 E。当 0.00 由 OFF→ON→OFF 时, 程序的执行顺序是 A→B→E; 当 0.01 由 OFF→ON→OFF 时, 程序的执行顺序是 C→D→E。其执行过程是:

对 A→B→E 分支, 当 0.00 由 OFF→ON→OFF 时, 执行指令 SNXT(009) H0.00, 启动并执行步 A, 使 100.00 为 ON; 在 0.02 由 OFF→ON→OFF 时, 结束、复位步 A 并启动步 B, 使 100.00 为 OFF、100.01 为 ON; 在 0.03 由 OFF→ON→OFF 时, 结束、复位步 B 并启动步 E, 使 100.01 为 OFF、100.00 为 ON; 当 0.06 由 OFF→ON→OFF 时, 结束、复位步 E, 使 100.00 OFF, 该分支步程序结束。

对 C→D→E 分支, 当 0.01 由 OFF→ON→OFF 时, 执行指令 SNXT(009) H0.02, 启动并执行步 C, 使 100.02 为 ON; 在 0.04 由 OFF→ON→OFF 时, 结束、复位步 C 并启动步 D, 使 100.02 为 OFF、100.01 为 ON。在 0.05 由 OFF→ON→OFF 时, 结束、复位步 D 并启动步 E, 使 100.01 为 OFF、100.00 为 ON。当 0.06 由 OFF→ON→OFF 时, 结束、复位步 E, 使 100.00 为 OFF, 该分支步程序结束。

本例是选择分支步程序, 所以两个分支中首步的启动条件互锁。

程序中, 以 0.05 为执行条件的指令 SNXT(009) H0.04 与 STEP(008) H0.04 相邻, 这与前面介绍的步进程序的结构形式相符。而以 0.03 为执行条件的指令 SNXT(009) H0.04 与 STEP(008) H0.04 相隔较远, 这种编写法是允许的。

#### 3. 并行分支执行类步进程序的执行过程

图 3.99 是并行分支类的步进程序。图 3.99(a) 是图 3.99(b) 的流程图。从流程图可以看出, 这是由两个分支的步进程序和一段普通程序组成的程序段。步进程序中每个分支各有两个步。当 0.00 由 OFF→ON→OFF 时步程序启动, 分支 A→B 与分支 C→D 同时开始执行。当 0.03 由 OFF→ON→OFF 时, 两分支同时进入步 E。其执行过程如下:



图 3.97 步进程序基本的结构

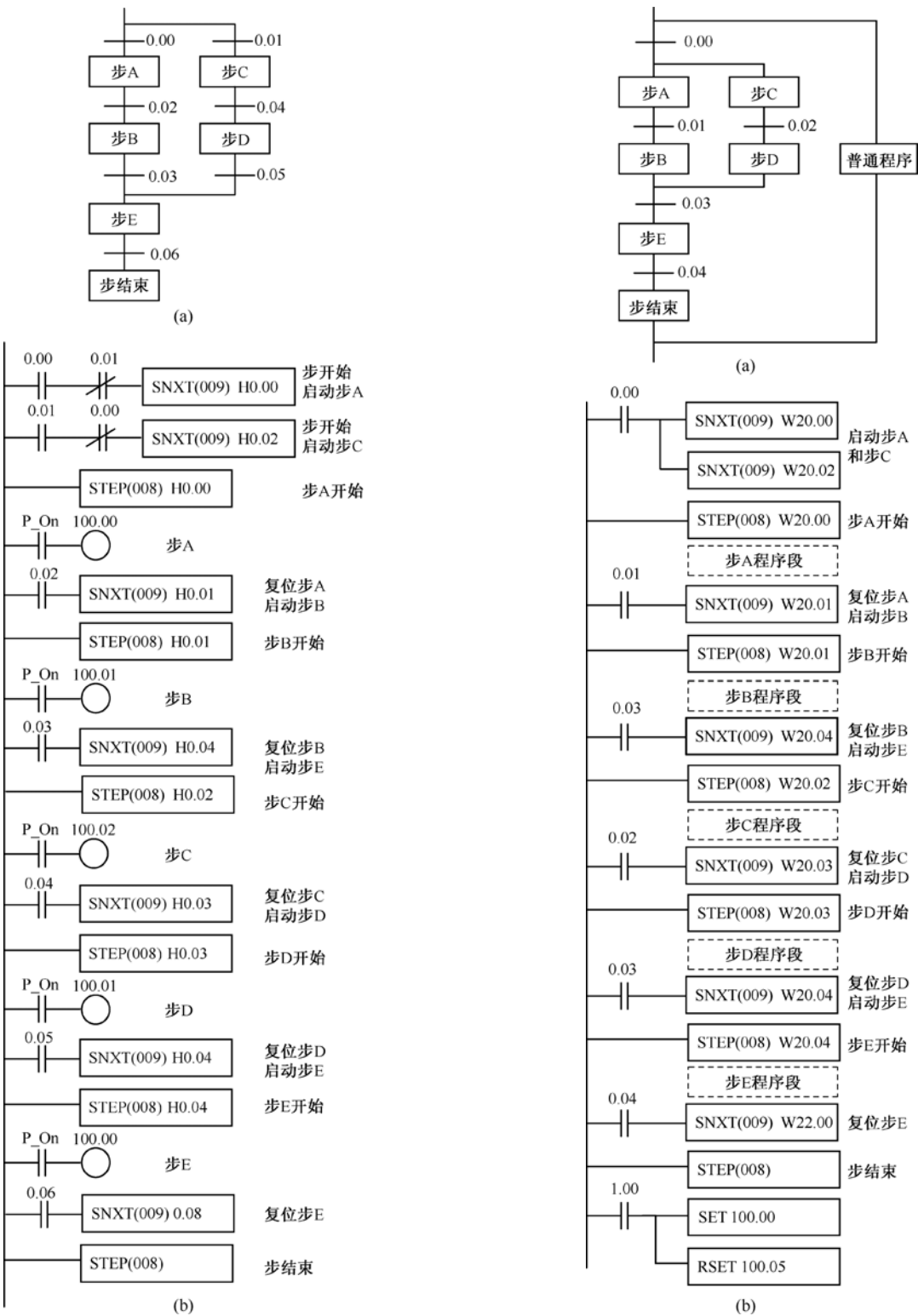


图 3.98 选择分支执行类的步进程序

图 3.99 并行分支执行类的步进程序

普通程序的执行不受步程序的制约。在 PLC 加电后,无论 0.00~0.04 是何种状态,100.00 和 100.05 的状态只取决于 1.00。

当 0.00 由 OFF→ON→OFF 时,执行指令 SNXT(009) W20.00 和 SNXT(009) W20.02,同时启动并执行步 A 和步 C。对分支 A→B,在 0.01 由 OFF→ON→OFF 时,结束、复位步 A 并启动步 B;对分支 C→D,在 0.02 由 OFF→ON→OFF N 时,结束、复位步 C 并启动步 D。当 0.03 由 OFF→ON→OFF 时,结束、复位步 B 和步 D,启动并执行步 E。在 0.04 OFF→ON→OFF 时,结束、复位步 E,步程序结束。

程序中,以 0.03 为执行条件的指令 SNXT(009) W20.04 在两处出现过。一个用来清除步 B、启动步 E 的,另一个用来清除步 D、启动步 E 的。

本例中,在步进程序段的末尾安排了一段普通程序,其执行不受步程序的制约。同样在步进程序段之前也可以安排普通程序。

与用其他指令编写的、有相同功能的程序相比,用步进控制指令编写的程序语句比较多,但是这种程序的优点是逻辑关系清楚,程序编写过程中不易出差错,而且程序可读性好。

3.15.4 步进控制实例

例 3-15-1 步进指令示例,如图3.100 所示。

当 0.00 为 ON 时执行 W0.00 程序段;当 0.01 为 ON 时执行 W0.01 程序段,此时被 W0.00 程序段使用过的数据区的状态如下:

- ① 用 OUT、OUT NOT 指令指定的继电器为 OFF;
- ② 定时器复位;
- ③ 其他保持。

例 3-15-2 单步启动标志 A200.12 的应用。

CPU 执行 STEP 指令时,单步启动标志 A200.12 仅 ON 1 个扫描周期。利用 A200.12 复位计数器,如图 3.101 所示。

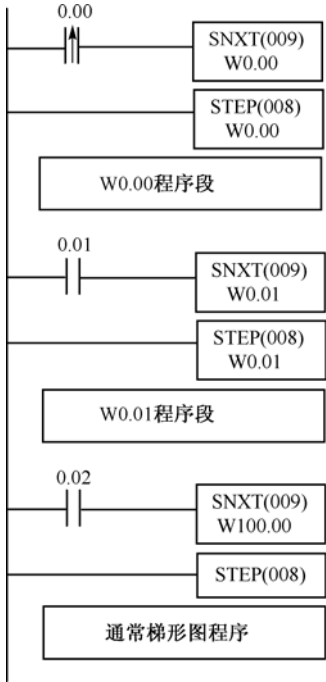


图 3.100 例 3-15-1 图

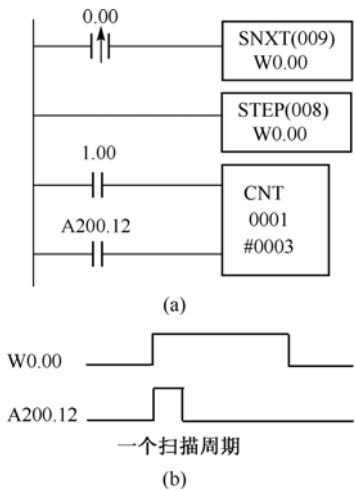


图 3.101 单步启动标志 A200.12 应用示例图

例 3-15-3 某零件的装配过程按上料、组装和分检三个工序顺序实施，如图3.102所示。各工序由传送带旁的传感器(SW1~SW4)发出信号，驱动对应机构动作，后者每完成一次操作都要回复原位，等待下一个信号。此过程的顺序流程如图3.103所示。

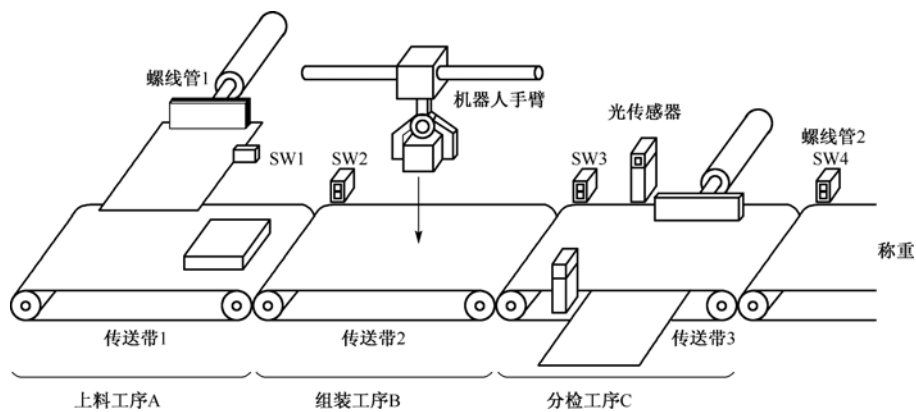


图 3.102 装配过程示意图

此顺序装配过程的控制梯形图采用了步进指令，如图 3.104 所示，每个不同的 SNXT 指令与步程序一一对应，而来自现场的传感器信号将启动对应的步程序。

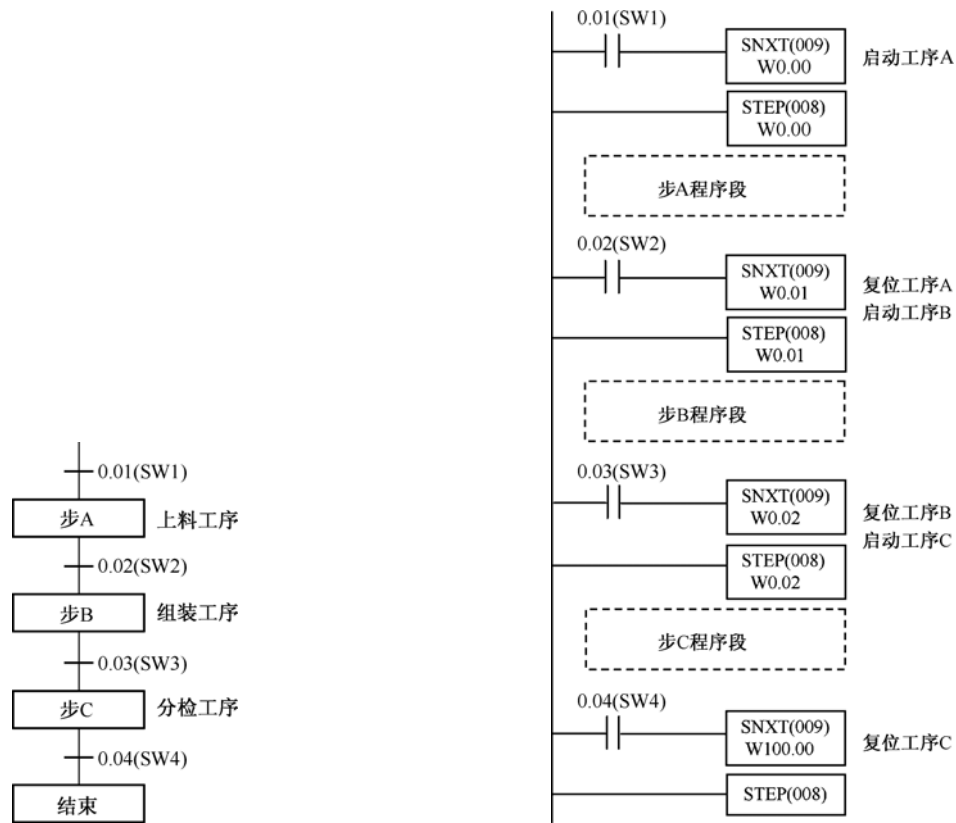


图 3.103 顺序流程图

图 3.104 顺序控制梯形图

例 3-15-4 选择控制。  
某产品按质量分选后打印标签，如图3.105所示，产品将被 A 或 B 传送带输送，传感器 SWA1 和

SWB1 发出信号，驱动对应的传送带 A 或 B 运行，最终经打印机打印标签。此选择过程的顺序流程如图3.106所示，梯形图程序如图3.107所示。

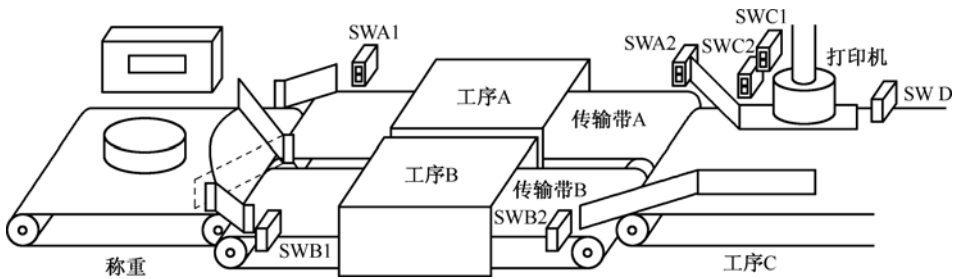


图 3.105 分选过程示意图

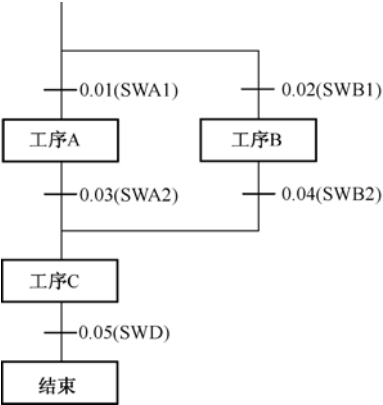


图 3.106 顺序流程图

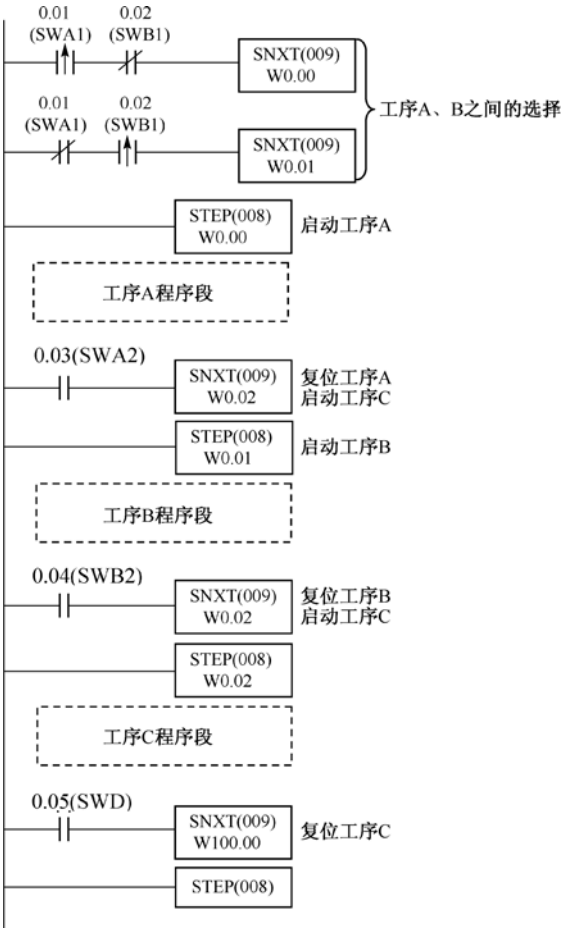


图 3.107 选择控制梯形图

例 3-15-5 并行控制。

某产品组装流程如图3.108所示，两个部件要分别经过工序 A、B 和工序 C、D 处理后才能在工序 E 组装成产品，位置传感器 SW1~SW7 指示各工序的工作状态。

此并行组装过程的顺序流程图如图3.109所示，SW1~SW2 同时启动工序 A、C，工序 A 完成后运行工序 B，工序 C 完成后运行工序 D。工序 B 和 D 都完成时，才能运行工序 E。梯形图控制程序如图3.110所示。

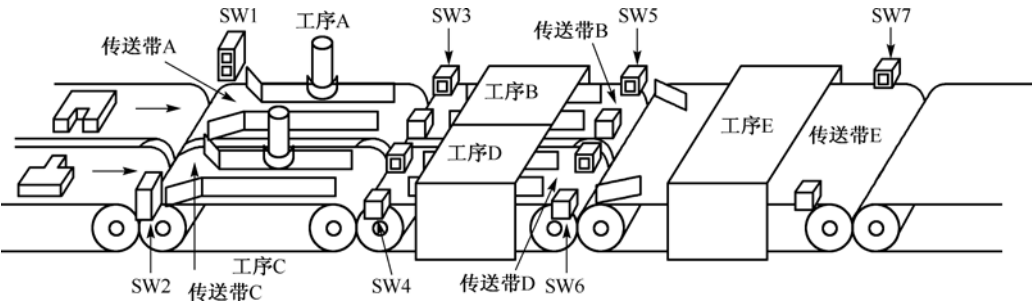


图 3.108 并行组装过程示意图

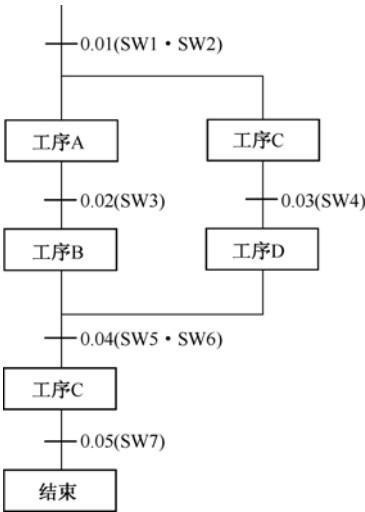


图 3.109 顺序流程图

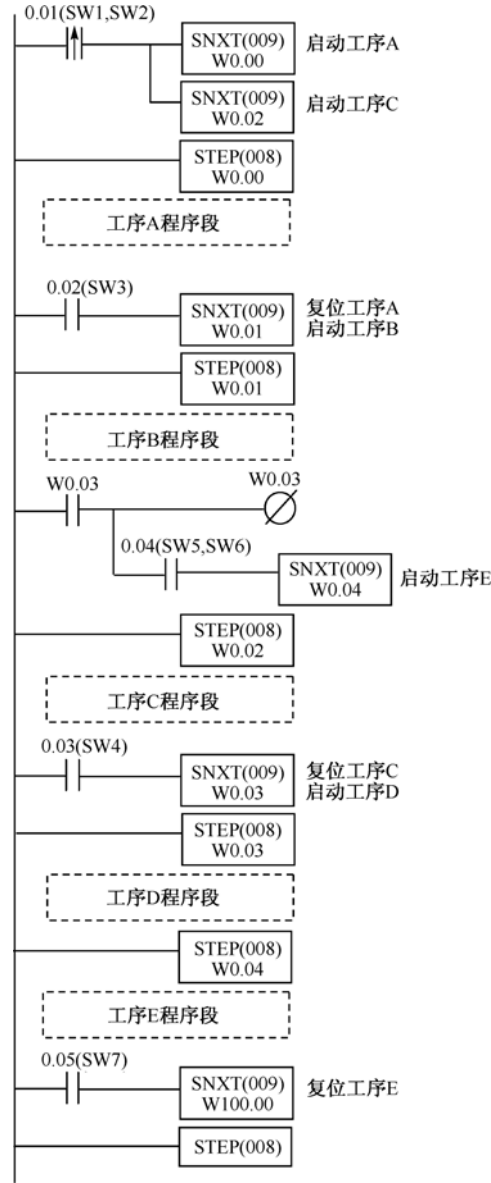


图 3.110 选择控制梯形图

## 习 题

1. 一条指令通常由哪几部分组成？各部分的作用是什么？
2. CPIH 的指令共有几种形式？执行微分型指令和非微分型指令时有什么区别？什么情况下要使用微分型指令？微分型指令的功能还可以通过何种方法实现？
3. 分别画出图 3.111 所示两个语句表的梯形图。
4. 写出图 3.112 所示梯形图的语句表，并画出 100.00 的工作波形。

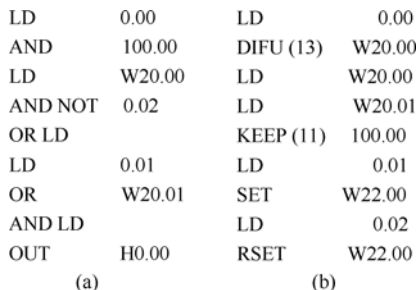


图 3.111

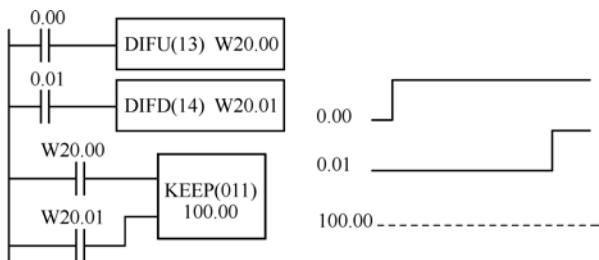


图 3.112

5. 把图 3.113(a)、(b)改画成能节省语句的形式，将图3.113(c)按 PLC 梯形图的规则进行转换。

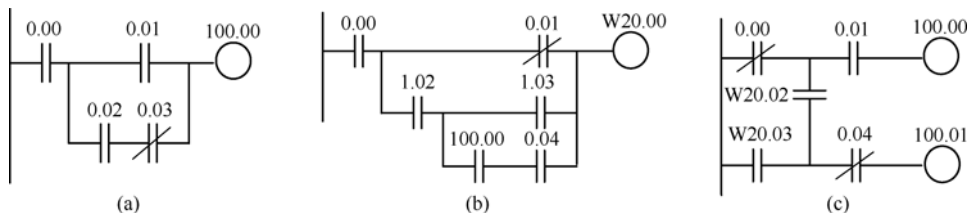


图 3.113

6. 试设计一个电动机正反转自动控制程序，要求电动机启动后反转 8 s 后停 15 s，再正转 8 s，停 15 s，如此循环，直到按下停止按钮。编程中需要考虑互锁。
7. 编一个通断均为 1 s 的方波发生器程序，设输入 0.02 来启动发生器，方波由继电器 100.02 输出。画出梯形图，写出语句表。
8. 用 TIM 指令编写一个程序，实现控制：在 0.00 接通后 10 s 100.00 接通并保持。100.00 接通 10 s 后自动断开。画出梯形图，写出语句表。
9. 用 CNT 和 TIM 指令分别来实现开关 0.00 接通 15 s 后，100.00 接通；0.00 断开后，100.00 经过 5 s 后再断开的功能，设计梯形图程序。
10. 写出图 3.114(a)的语句表，画出图 3.114(b)中 100.00、100.01 和 W200.00 的波形图。
11. 写出图 3.115 所示梯形图的语句表，并画出 100.00 和 100.01 的波形图。
12. 按下面的要求，用 JMP/JME 指令编写一个程序：  
当闭合控制开关时，灯 1 和灯 2 亮，经过 10 s 两灯均灭。当断开控制开关时，灯 3 和灯 4 开始闪烁(亮 1 s、灭 1 s)。经过 10 s 两灯均灭。
13. 设计一个 24 小时的计时程序，要求每小时产生一个 2 s 的外部输出(设输出口为 100.05)，每 24 小时产生一个 10 s 的外部输出(设输出口为 100.06)，用输入 0.05 来启动这个程序。

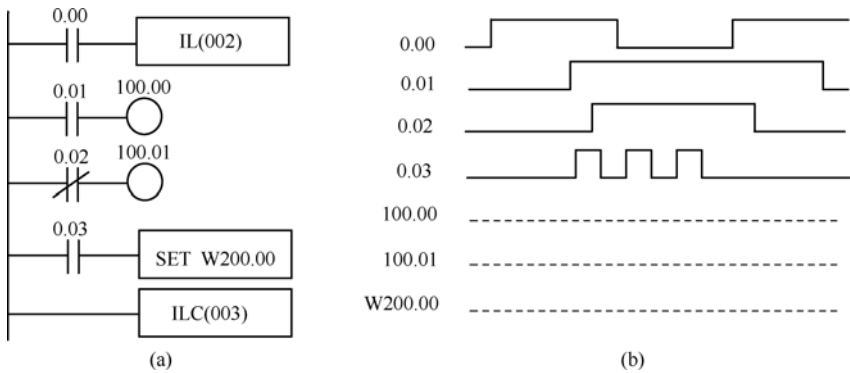


图 3.114

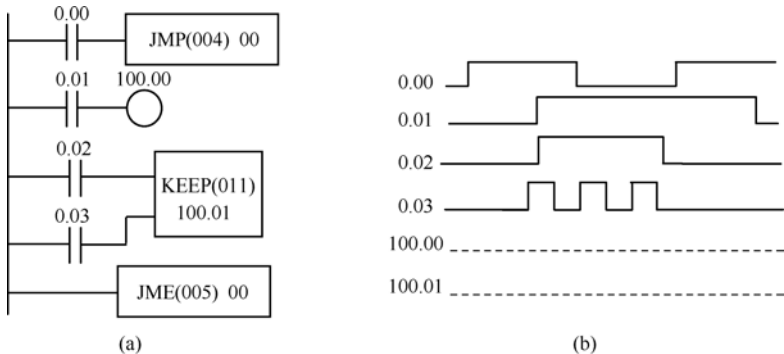


图 3.115

14. 分别用 CNTR 的加、减计数指令编写一个能记录 1 万个计数脉冲的循环计数器程序，画出梯形图，写出语句表。
15. 写出图 3.116 所示梯形图的语句表，并画出各指定的波形图。
16. 在图 3.117 所示梯形图中，当 0.00 为 OFF、0.01 为 ON 时为连续移位；当 0.01 为 OFF 时，用 0.00 进行手动移位。试分析两种情况下移位过程的差别，并画出连续移位时 W200 通道相关位的波形。

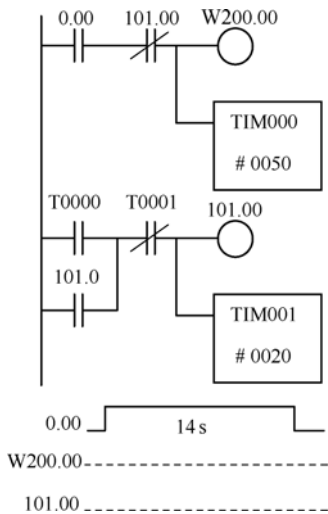


图 3.116

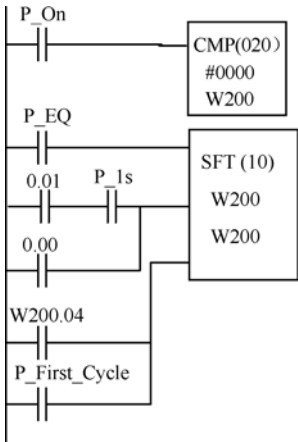


图 3.117



17. 按如下要求设计一个程序，画出梯形图，写出语句表。

- (1) 在 PLC 加电的第一个扫描周期，计数器能自动复位，当计数器达到设定值时也能自动复位；
- (2) CNT 的设定值为#1500，每隔 0.1 s 其当前值减 1；
- (3) 用 MOV 指令将#1000 传送到通道 W210；
- (4) 将通道 W210 的内容与 CNT 比较，若通道 210 的内容 < CNT 的当前值，100.00 为 ON，若通道 W210 的内容 > CNT 的当前值，100.01 为 ON，若通道 W210 的内容 = CNT 的当前值，100.02 为 ON。

18. 按下面的要求，用 MOV、CNT、BSET 指令设计一个程序。画出梯形图，写出语句表。

- (1) 将计数器的设定值#0500 传送到 H0 中。
- (2) 0.00 每接通一次，计数器的当前值减 1，计数器达到设定值时能自复位。
- (3) 计数过程中随时可以改变计数器的当前值。

19. 画出图 3.118 所示梯形图中 100 通道各输出位的波形图。

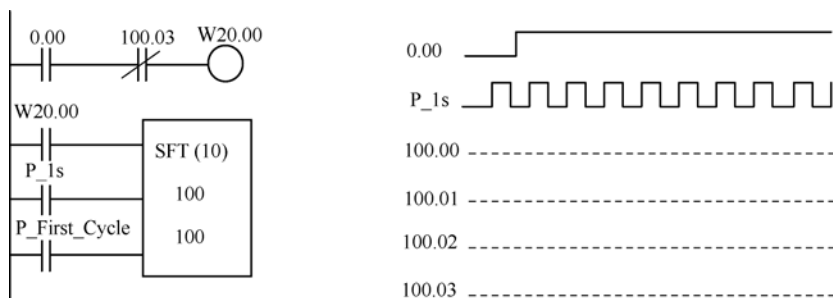


图 3.118

20. 用 W200 通道做 SFTR 指令的控制位，设计一个可逆移位寄存器程序。当 0.00 为 ON、0.01 为 OFF 时，100 通道最低位的“1”每秒左移 1 位；当 0.01 为 ON、0.00 为 OFF 时，100 通道最高位的“1”每秒右移 1 位。画出梯形图，写出语句表。

建议：可配合使用 KEEP 指令和 P\_1s。

21. 请完成下列各题。

(1) 指令 BIN(023)的操作数 S 为 W220(内容为 0318)，R 为 H10。执行一次该指令，请写出结果通道的内容。

(2) 指令 BCD(024)的操作数 S 为 W220(内容为 010E)，R 为 H10。执行一次该指令，请写出结果通道的内容。

(3) 指令 MLPX(076)的操作数 S 为 W220(内容为 E563)，C=#0023，R 的通道自定。执行一次该指令，请写出结果通道的内容。

(4) 指令 DMPX(077)的操作数 S 为 W220(W220 中为 0C1A、W221 中的为 182D)，R 的通道自定，C=#0013。执行一次该指令，请写出结果通道的内容。

22. 求  $(D10 + D11) \times D12$  的值。将结果放在 D13、D14 中，设 D10~D12 通道内的数均为 BCD 码，并做溢出判断。

23. 求  $(D10 + D11) \times D12$  的值。将结果放在 D13、D14 中，设 D10~D12 通道内的数均为十六进制数，并做溢出判断。

24. 通道 H10 中的内容是 10AB。用 ASC 指令将通道 H10 中的第 0 位数字和第 1 位数字转换成 ASCII 码，结果放在 D0 开始的通道中。要求进行偶校验，转换结果从 D0 通道的高 8 位开始存放。画出梯形图，写出语句表，并画出转换前后 H10 和 D0 等通道数据的对应关系。

25. 分别编写一个程序完成下列各运算，并画出梯形图，写出语句表。
- (1) H1 的内容为#3210，H0 的内容为#7601。用+BC(406)指令完成(3210+7601)的运算，结果放在 D0 中，进位放在 D1 中。
- (2) H1 的内容为#3210，H0 的内容为#7601。用-BC(416)指令完成(3210-7601)的运算，结果放在 D0 中，进位放在 D1 中。
- (3) 用十进制运算指令编写一个程序，完成(200-100)×2/10 的运算，运算结果放在 D 数据区。
26. 用二进制运算指令编写一个程序，完成[(250×8+200)-1000]/5 的运算，运算结果放在 D 数据区。请画出梯形图，写出语句表。
27. 用逻辑运算指令分别编写一个程序以实现下列各要求，结果放在 D 数据区。请画出梯形图，写出语句表。
- (1) 将 W200 通道全清零。
- (2) 保留 W200 通道中低 8 位的状态，其余为 0。
- (3) 令 W200 通道中是 1 的位变为 0，是 0 的位变为 1。
- (4) 将 W200 通道各位全置为 1。
28. 程序如图3.119、图3.120所示，指出下面程序带符号数的运算结果，即说明结果通道的内容。设 D100 内容为#0187，D200 内容为#F8C5；H 100 内容为#0187，H200 内容为#F8C5。

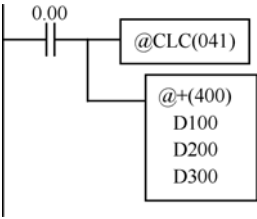


图 3.119

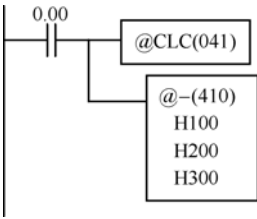


图 3.120

29. 用浮点数运算指令编写梯形图程序，求下列算式的结果：
- $$y = \frac{1.32^2 - \sqrt{18.9}}{\sin(\pi/3)} \times (-0.885)$$
30. 用子程序控制指令分别编写一个能实现下列控制要求的程序。请画出梯形图，写出语句表。
- (1) 某系统中，当温度传感器发出信号时，A、B 两台电动机就按下面的规律运行一次：A 电动机运行 5 min 后，B 电动机启动并运行 3 min 后停车；A 电动机在运行 10 min 时自行停车。
- (2) 两个计数器分别记录两个加工站的产品数量。每过 15 min 要进行一次产品数量的累计(设 15 min 内每个加工站的产品数量都不超过 100)，经过 8 小时计数器停止计数。

## 第 4 章 编程软件 CX-P

欧姆龙大部分 PLC 的编程可以通过简易编程器或计算机辅助编程来实现。但本书所介绍的 CP1 系列 PLC 只能通过计算机辅助编程。计算机辅助编程既省时省力,又便于程序管理,它具有简易编程器无法比拟的优越性,是一种广泛应用的编程方式。计算机辅助编程时要安装专用编程软件,还要和 PLC 建立通信连接。PLC 与计算机通信时,通常使用 CPU 单元内置的通信口。通信口大多为 RS-232C 口,有时也用 RS-422 口。CP1H 内置了 USB 通信口,可以方便地与带 USB 总线的计算机通信。

欧姆龙使用的编程软件是 CX-Programmer,简称 CX-P。使用编程软件可以实现的功能有:梯形图或语句表编程,编译检查程序,数据和程序的上载、下载及比较,对 PLC 的设定区进行设置,对 PLC 的运行状态及内存数据进行监控和测试,打印程序清单等文档,管理文件等。

欧姆龙目前流行的 CX-One 软件包是一个集成的编程和配置环境。CX-One 包括 OMRON 全系列产品开发软件,有网络软件、PLC 软件、HMI(Human Machine Interface)软件、运动&驱动软件和过程控制软件,其中 PLC 软件含有 CX-Programmer、CX-Simulator 和 SwitchBox Utility。CX-One 集成软件大大减少了配置工作的复杂性。如果手中有 CX-One,安装时只选择 CX-Programmer 进行安装即可进行编程。

### 4.1 CX-P 简介

CX-P 编程软件有多个版本,从 2000 年的 CX-P 2.0 开始,发展到 2010 年的 CX-P 9.2,高版本兼容低版本的功能。它支持 C、CV/CVM1、CS1、CJ1、CP1 等 OMRON 全系列的 PLC。CX-P 具有强大的编程、调试、监控功能和完善的维护功能,使程序开发及系统维护更为简单、快捷。

CX-P 最新的版本为 9.2,主要特性有:

- ① 以树状目录的形式分层显示一个工程的各个项目,这些项目能够被直接访问。
- ② 具有 Windows 风格的界面,可以使用菜单、工具栏和键盘快捷键操作。用户可自定义工具栏和快捷键。鼠标可以使用拖放功能,使用右键显示快捷菜单进行各种操作等。
- ③ 在单个工程下支持多个 PLC,一台计算机可与多个 PLC 建立在线连接,支持在线编程;单个 PLC 下支持一个应用程序,其中, CV/CVM1、CS1、CJ1、CP1 系列的 PLC 可支持多个应用程序(任务);单个应用程序下支持多个程序段,一个应用程序可以分为一些可自行定义的、有名字的程序段,因此能够方便地管理大型程序。可以一人同时编写、调试多个 PLC 的程序;也可以多个人同时编写、调试同一 PLC 的多个应用程序。
- ④ 提供全清 PLC 内存区的操作。对 PLC 进行初始化操作,清除 CPU 单元的内存,包括用户程序、参数设定区和 I/O 内存区。
- ⑤ 可对 PLC 进行设定,例如, CX-P 对 CP1H,可进行“启动”、“循环时间”、“中断/刷新”、“错误设定”、“外围端口”和“高速计数器”等的设定,设定下载至 PLC 后生效。
- ⑥ 支持梯形图、语句表、功能块、结构文本编程和顺序功能图编程。梯形图和语句表是最常用的编程语言,OMRON 所有 PLC 都支持梯形图和语句表编程。除此之外,OMRON 的 CS1、CJ1、CP1 等主流的 PLC 还可用功能块、结构文本语言编程以及顺序功能图编程。
- ⑦ CX-P 除了可以直接采用地址和数据编程外,还提供了符号编程的功能,编程时使用符号而不必考虑其位和地址的分配。符号编程使程序易于移植和拖放。

⑧ 颜色的使用。颜色的使用可以自定义。默认设置时，全局和本地符号在梯形图中具有不同的颜色，梯形图中的错误显示为红色。

⑨ 程序可分割显示以监控多个位置。一个程序能够在垂直和水平分开的屏幕上被显示，可同时显示在 4 个区域上，这样可以监控整个程序，同时也监控或输入特定的指令。

⑩ 提供丰富的在线监控功能，方便程序调试。为了检查程序的逻辑性，监视可以暂时被冻结等。CX-P 与 PLC 在线连接后，可以对 PLC 进行各种监控操作，例如，置位/复位，修改定时器/计数器设定值，改变定时器/计数器的当前值，以十进制、有符号的十进制、二进制或十六进制的形式观察通道内容，修改通道内容，计算扫描周期等。

⑪ 可对 PLC 设计 I/O 表，为 PLC 系统配置各种单元(板)，并对其中的 CPU 总线单元和特殊 I/O 单元设定参数。I/O 表设计完成后要下载到 PLC 中进行登记，一经 I/O 表登记，PLC 运行前将检查其实际单元(板)与 I/O 表是否相符，如不符，PLC 不能运行，这样可避免出现意外情况。

⑫ 通过 CX-Server 软件的应用，可以使 PLC 与它支持的各类网络进行通信，使用 CX-Server 中的网络配置工具 CX-Net 可以设置数据链接表和路由表(关于 CX-Server 的使用，请查阅 CX-Server 手册)。

⑬ 具有远程编程和监控功能。上位机通过被连接的 PLC 可以访问本地网络或远程网络的 PLC。上位机还可以通过 MODEM，利用电话线访问远程 PLC。


⑭ 可对 PLC 程序进行加密。OMRON CV/CVM1、CS1、CJ1、CP1 系列 PLC 可以用 CX-P 设置密码，需在程序的开头编一小段包含密码的程序，密码为 8 位字母或数字。

⑮ 可以修改输入模式，在“工具”菜单的“更改输入模式”中，可选择智能输入模式或经典模式。智能输入模式支持自动显示建议的指令和地址。

CX-P 软件的功能强大，操作内容很多，不可能全部介绍，在本章后续内容中，仅围绕编程常见应用问题加以介绍，更详细的说明请参见 CX-P 使用手册。

## 4.2 CX-P 的使用

### 4.2.1 CX-P 主窗口

- ① 启动 CX-P 后，窗口显示如图4.1 所示。
- ② 单击“文件”菜单中的“新建”命令或快捷按钮，出现如图4.2所示的“变更 PLC”对话框。

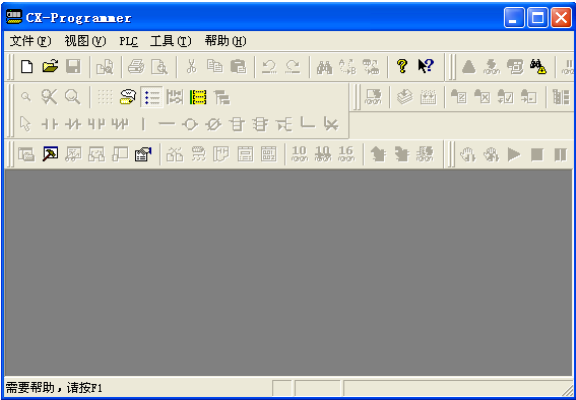


图 4.1 CX-P 窗口

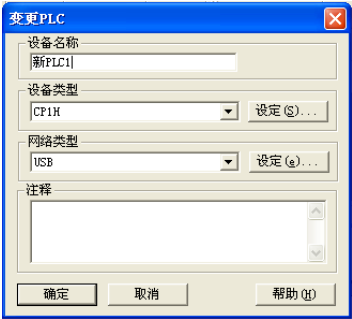


图 4.2 “变更 PLC”对话框

- ③ 在“设备名称”文本框中输入用户为 PLC 定义的名称，例如输入“TrafficController”。

- ④ 在“设备类型”下拉列表框中选择 PLC 的系列，例如选择“CPIH”。单击“设定”按钮可进一步配置 CPU 型号，例如选择“XA”。
- ⑤ 在“网络类型”下拉列表框中选择 PLC 的网络类型，如选择“USB”。
- ⑥ 在“注释”编辑框中输入与此 PLC 相关的注释。
- ⑦ 在图4.2“变更 PLC”对话框中，单击“确定”按钮，显示如图4.3所示 CX-P 主窗口，表明建立了一个新工程。若单击“取消”按钮，则放弃操作。

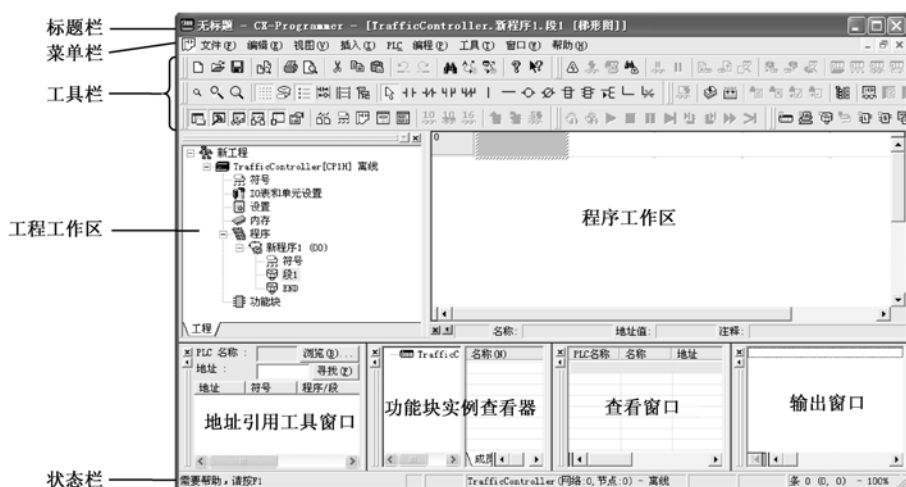




图 4.3 CX-P 主窗口

CX-P 主窗口的组成如下。

- ① 标题栏：显示打开的工程文件名称、编程软件名称和其他信息。
  - ② 菜单栏：将 CX-P 的全部功能，按各种不同的用途组合起来，以菜单的形式显示。通过单击主菜单及子菜单中的命令，可进行相应的操作。
  - ③ 工具栏：将 CX-P 中经常使用的功能以按钮的形式集中显示，工具栏内的按钮是执行各种操作的快捷方式之一。可以通过“视图”菜单中的“工具栏”来选择要显示的快捷按钮。
  - ④ 状态栏：位于窗口的底部，状态栏显示即时帮助、PLC 在线/离线状态、PLC 工作模式、连接的 PLC 和 CPU 类型、PLC 扫描循环时间、在线编辑缓冲区大小，以及显示光标在程序窗口中的位置。可以通过“视图”菜单中的“状态栏”命令来打开和关闭状态栏。
  - ⑤ 工程工作区：位于主窗口的左边，显示一个工程的分层树形结构。一个工程下可生成多个 PLC，每一个 PLC 包括符号、IO 表和单元设置、设置、内存、程序及功能块等，而每一个程序包括本地符号表和程序段。
  - ⑥ 程序工作区：位于主窗口的右边，是编辑梯形图和助记符程序的区域。当建好一个新的工程或者把一个新的 PLC 添加到工程中时，程序工作区将显示一个空的窗口。
  - ⑦ 其他窗口：输出窗口、查看窗口、功能块实例查看器和地址引用工具窗口。
- 输出窗口：位于主窗口的下面，可以显示编译程序结果、查找报表和程序传送结果等。
- 查看窗口：用于在线状态下监视梯形图中某继电器的数值变化。
- 功能块实例查看器：查看功能块的工作情况。
- 地址引用工具窗口：用于检索大型梯形图中具有相同地址号的继电器位。

4.2.2 CX-P 工程

在工程工作区，工程中的项目以分层树形结构显示，如图4.4所示，分层树形结构可以通过  压缩或者扩展。工程中的每一个项目都有图标相对应，图 4.4 为在线状态下的显示，离线状态下不会显示“错误日志”和“PLC 时钟”。

对工程中的某一项目进行操作时，可以右击该项目的图标，出现快捷菜单后，选择相应的命令；双击某一项目，可以直接打开该项目所对应的窗口。

下面介绍工程的各个项目及相关操作。



图 4.4 工程中的项目

1. 工程

对工程进行的操作有插入 PLC、粘贴、重命名、属性等。

2. PLC

对 PLC 进行的操作有修改、插入程序，在线工作、在线模拟，(改变 PLC)操作模式、监视，自动分配、编译所有的 PLC 程序，剪切、复制、粘贴、删除，属性等。

3. 全局符号表和本地符号表

(1) 符号

CX-P 除了直接采用地址和数据编程外，还提供了符号编程的功能。符号是用来表示地址、数据的标识符。一个 PLC 下各个程序都可以使用的符号称为全局符号，为某个程序定义的专有的符号称为本地符号。

在编程中使用符号，具有简化编程、增强程序可读性、方便程序维护等优点。例如，仅改变一下符号对应的地址，程序就会自动使用新地址。程序越复杂，符号编程的优势越显著。

符号除了分配地址或数值外，还要规定数据类型。符号的数据类型见表 4.1。

表 4.1 符号的数据类型

符号名称	容量	符号	格式	备注
BOOL	1 位	—	二进制	逻辑二进制地址位，用于接触点和线圈
CHANNEL	1 个或多个字	—	任意	任何除 BOOL 和 NUMBER 的非位地址
DINT	2 个字	有	二进制	一个有符号的双字二进制字地址
INT	1 个字	有	二进制	一个有符号的单字二进制字地址
LINT	4 个字	有	二进制	一个有符号的四字二进制字地址
NUMBER	—	有	十进制	是一个数字值，而不是一个地址。这个值可以有符号数或者浮点数，默认时为十进制，可以使用前缀“#”来表明它是一个十六进制数
REAL	2 个字	有	IEEE	一个双字浮点值的地址
LREAL	4 个字	有	IEEE	一个四字浮点值的地址
UDINT	2 个字	无	二进制	一个无符号的双字二进制字地址
UDINT_BCD	2 个字	无	BCD	一个无符号的双字 BCD 地址
UINT	1 个字	无	二进制	一个无符号的单字二进制字地址
UINT_BCD	1 个字	无	BCD	一个无符号的单字 BCD 地址
ULINT	4 个字	无	二进制	一个无符号的四字二进制字地址
ULINT_BCD	4 个字	无	BCD	一个无符号的四字 BCD 地址

对 PLC 的定时器/计数器，使用 BOOL 数据类型来定义其定时/计数完成标志(触点)，例如，将“T0001”定义为 BOOL 类型的符号“RTimerDone”，RTimerDone 代表 TIM0001 的定时完成标志(触点)；使用 NUMBER 数据类型来定义定时器号和设定值，例如，将“TIM0001”的“0001”定义为 NUMBER 类型的符号“RTimer”，将设定值定义为 NUMBER 类型的符号“TimeInterval”。

由于规定了符号的数据类型，CX-P 能够检查符号是不是以正确的方式被使用。例如，一个符号定义为 UINT\_BCD 类型，这表示其代表的数据是无符号 BCD 单字整数。CX-P 对该符号进行检验时，能检查出其是否只被使用于操作数是 BCD 类型的指令，如果不是，则给出警告。

(2) 全局符号表和本地符号表

符号表是一个可以编辑的符号列表，包括名称、数据类型、地址/值、注释等。对 CJ1 系列和 CS1 系列的 PLC，这个列表还提供关于机架位置等信息。每一个 PLC 下有一个全局符号表，当工程中添加了一个新 PLC 时，根据 PLC 型号的不同，全局符号表中会自动添入一些预先定义好的与该型号有关的符号。每一个程序下有一个本地符号表，其包含只有在这个程序中要用到的符号，本地符号表被创建时是空的。

在符号表中，每一个符号名称在表内必须是唯一的。但是，允许在全局符号表和本地符号表里出现同样的符号名称，这种情况下，本地符号优先于同样名称的全局符号。

双击“全局符号表”图标，可以显示出全局符号表，如图4.5所示。全局符号表中最初自动添加的一些预置的符号取决于 PLC 类型。例如，许多 PLC 都能生成的符号“P\_1s”（1.0 秒的时钟脉冲位）。所有的预置符号都具有前缀“P\_”，其不能被删除或者编辑。



名称	数据类型	地址 / 值	机架位置	使用	注释
▾ P_0_02s	BOOL	CF103		工作	0.02秒时钟脉冲位
▾ P_0_1s	BOOL	CF100		工作	0.1秒时钟脉冲位
▾ P_0_2s	BOOL	CF101		工作	0.2秒时钟脉冲位
▾ P_1s	BOOL	CF102		工作	1.0秒时钟脉冲位
▾ P_1分钟	BOOL	CF104		工作	1分钟时钟脉冲位
▾ P_AER	BOOL	CF011		工作	访问错误标志
▾ P_CIO	WORD	A450		工作	CIO区参数
▾ P_CY	BOOL	CF004		工作	进位(CY)标志
▾ P_Cycle_Time_Error	BOOL	A401.08		工作	循环时间错误标志
▾ P_Cycle_Time_Value	UINT	A264		工作	当前扫描时间
▾ P_DM	WORD	A460		工作	DM区参数
▾ P_EM0	WORD	A461		工作	EM0区参数
▾ P_EM1	WORD	A462		工作	EM1区参数
▾ P_EM2	WORD	A463		工作	EM2区参数
▾ P_EM3	WORD	A464		工作	EM3区参数

图 4.5 全局符号表

双击“本地符号表”图标将显示出如图4.6所示的本地符号表。



名称	数据类型	地址 / 值	机架位置	使用	注释
▾ RTimer	NUMBER	0001			红灯定时器
▾ RTimerDone	BOOL	T0001		工作	红灯定时完成标志(触点)
▾ RTimer	NUMBER	0002			红黄灯定时器
▾ RTimerDone	BOOL	T0002		工作	红黄灯定时完成标志(触点)
▾ GTimer	NUMBER	0003			绿灯定时器
▾ GTimerDone	BOOL	T0003		工作	绿灯定时完成标志(触点)
▾ YTimer	NUMBER	0004			黄灯定时器
▾ YTimerDone	BOOL	T0004		工作	黄灯定时完成标志(触点)
▾ RedLight	BOOL	100.00	主机架 ...	输出	停止
▾ YellowLight	BOOL	100.01	主机架 ...	输出	准备通行/停止
▾ GreenLight	BOOL	100.02	主机架 ...	输出	通行
▾ TimeInterval	NUMBER	50			定时时间

图 4.6 本地符号表

在符号表中可以对符号进行编辑、插入、剪切、复制、粘贴、删除和重命名等操作。符号显示可选择大图标、小图标、列表和详细内容 4 种方式。

4. IO 表和单元设置

模块式 PLC，如 CJ1 和 CS1 系列，使用之初要先进行 IO 表的设置。可双击工程工作区中的“IO 表和单元设置”图标，将弹出“PLC IO 表”窗口，如图4.7所示。在该窗口可按系统实际 IO 单元配置进行选择，从而完成 IO 单元设置。下载到 PLC 后，设置生效。

5. PLC 设置

各种机型的 PLC 都开辟了系统设定区，CX-P 通过“设置”图标来实现各种系统参数设定。例如，对 CPIH，双击“设置”图标，出现如图4.8所示的“PLC 设定”对话框，用图中的选项卡（有 15 个）对 CPIH 的各种系统参数进行设置。



图 4.7 “PLC IO 表”窗口

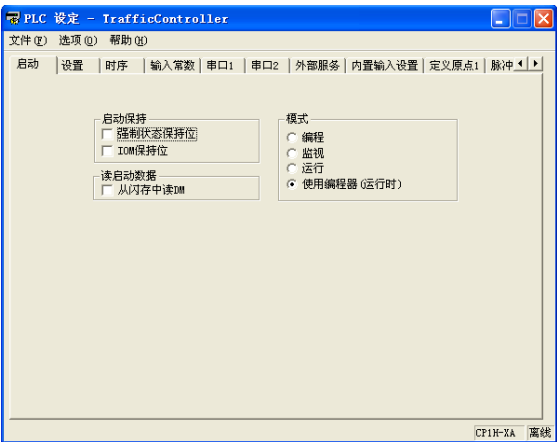


图 4.8 “PLC 设定”对话框

设置完毕，传送到 PLC 才能生效。

6. 错误日志

当 CPU 处于在线状态时，工程工作区的树形结构中将显示 PLC “错误日志”图标。双击该图标，出现如图4.9所示的“PLC 错误”窗口。窗口中有 3 个选项卡：“错误”、“错误日志”和“信息”。



图 4.9 “PLC 错误”窗口

- ① “错误”：显示 PLC 当前的错误状态。
- ② “错误日志”：显示有关 PLC 的错误历史。
- ③ “信息”：显示由程序设置的信息。信息可以被有选择地清除或全部清除。

7. PLC 内存

通过“内存”图标可以查看、编辑和监视 PLC 内存区，监视地址和符号、强制位地址，以及扫描和处理强制状态信息。

在工程工作区双击“内存”图标，将显示如图4.10 所示的“PLC 内存”窗口。  
左窗格的下方有两个选项卡：内存和地址选项卡。  
单击内存选项卡，在此选项卡中可完成如下操作。

- ① 数据的编辑。向 PLC 允许读/写操作的内存区输入和修改数据。输入的数据可选择的格式有二进制、BCD、十进制、有符号十进制、浮点、十六进制和文本。





图 4.10 “PLC 内存”窗口

② 数据的下载、上载及比较。下载是将计算机已编辑的 PLC 内存区数据传输到 PLC，上载是将 PLC 内存区的数据传输到计算机，比较是将计算机数据与 PLC 内存区比较。这三种操作必须在线状态下进行。

③ 数据的监视。在线状态下，监视 PLC 内存中某一数据区的数据变化。

④ 数据的清除和填充。在线状态下，可清除 PLC 内存区某一数据区的数据，或向某一数据区添加一个特定的值。输入的数据可选择的格式有二进制、BCD、十进制、有符号十进制、浮点、十六进制和文本。

单击地址选项卡，此选项卡包含“监视”和“强制状态”两个命令。在此选项卡中可完成如下的操作。

① “监视”命令。在线状态下，可通过该命令监视地址或符号、强制位地址。双击“监视”图标，出现地址监视窗口，在此窗口中输入一个地址或符号即可进行监视。当一个位正在被监视时，从该位的快捷菜单中，选中“强制”命令可对该位强制“ON”、“OFF”或“取消”强制状态。

② “强制状态”命令。在线状态下，可通过该命令，扫描和处理强制状态信息。双击“强制状态”图标，强制状态信息将显示在强制状态窗口中。选中某一强制状态位地址，从该位的快捷菜单中，可将其从强制状态窗口中复制到地址监视窗口中进行监视；可清除所有的强制位；可更新强制状态窗口。

## 8. PLC 程序

对 PLC 程序可以进行的操作有插入程序、剪切、复制、粘贴、删除、属性等。

## 9. 任务

目前 OMRON 生产的主流 PLC 有 CP1 系列、CJ1 系列和 CS1 系列，在程序上采用单元化结构，可以将程序按功能、控制对象、工序等条件进行划分，分割成“任务”的执行单位。

任务实际上是一段独立的具有特定功能的程序，每一个任务的最后一个指令应是 END，表示任务的结束。任务之间是独立的。任务可以单独上载或下载。

在工程目录中，“新程序 1”就是一个任务。任务由本地符号、段组成。最后一个段为 END，自动生成。

对任务进行的操作有打开、插入段，编译，部分传输，转移到条/步、注释条，剪切、复制、粘贴、删除，重命名，属性设置等。

10. 程序段

为了便于对大型程序的管理，可以将一个程序分成一些有定义、有名称的段，如图4.11所示，在程序名称下面显示了一个程序所包含的段的列表。一个程序可以分成多个段，如段 1、段 2 等，一个段就如同书的一章，PLC 按照顺序来搜索各段。程序中的段可以重新排序或重新命名。最后的段必须包含“END”指令。

在特定的程序中，可以使用段来储存经常使用的算法，这样就可以把段作为一个库，能够将其复制到另一个程序里面去。

对程序段进行的操作有打开梯形图、打开助记符，转移到条/步、注释条、剪切、复制、粘贴、删除，上移、下移，重命名，属性等。

可以直接用鼠标拖放一个段，在当前程序拖放将改变段的顺序，也可将段拖到另一个程序中。



图 4.11 程序段示意图

11. 功能块

OMRON 的 CS1、CJ1、CP1 等 PLC 可以使用功能块编程，可以从 OMRON 的标准功能块库文件或其他库文件中调入功能块，也可由用户使用梯形图或结构文本编辑产生。

4.2.3 CX-P 视图

图 4.12 显示了 CX- P 窗口的各种视图。



图 4.12 CX- P 窗口的各种视图

CX-P 的标准工具栏如图4.13所示，如新建、打开、保存等，此处不再赘述，按它们在 Word 中的用法使用即可。

如图4.14所示为查看工具栏的按钮，结合“窗口”菜单来控制图4.12中各种视图的显示和布局。



图 4.13 标准工具栏

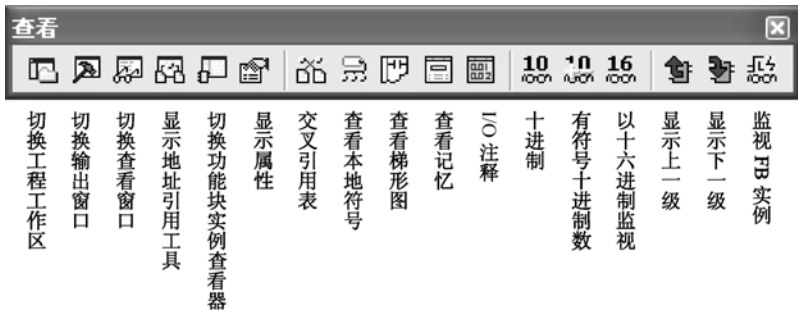


图 4.14 查看工具栏

1. 工程工作区

打开工程文件后，单击查看工具栏中的“切换工程工作区”按钮来激活此视图，再次单击该按钮将关闭此视图。对“工程工作区”中各个项目的介绍见 4.2.2 节。

2. 梯形图视图

双击工程工作区中的“段 1”图标，或选中“段 1”，单击工具栏中的“查看梯形图”按钮，在程序工作区将显示如图 4.15 所示梯形图视图。

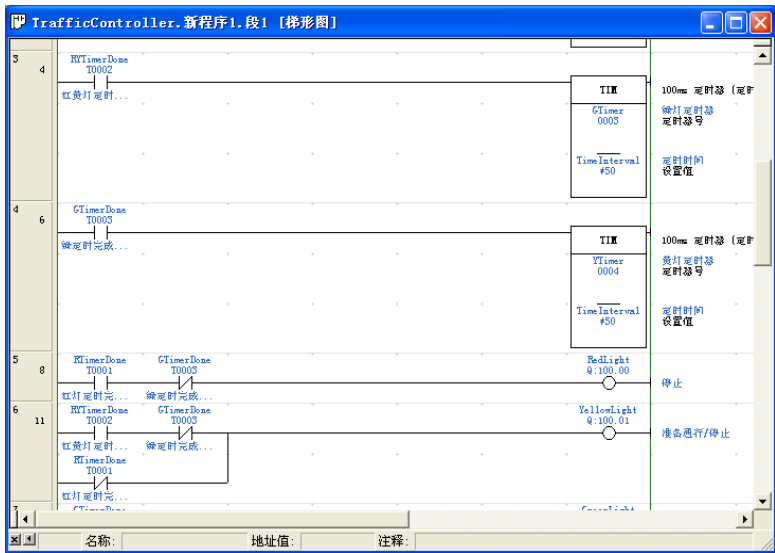


图 4.15 梯形图视图

(1) 梯形图视图的特征

梯形图视图的特征用以下名词描述。

- ① 光标：一个显示在梯级里面的当前位置的方块。光标的位置随时显示在状态栏中。
- ② 梯级 (条)：梯形图程序的一个逻辑单元。一个梯级能够包含多个行和列。所有的梯级都具有编号。

③ 梯级总线(母线): 左总线是指梯形图的起始母线, 每一个逻辑行必须从左总线画起。梯级的最右边是结束母线, 即右总线。右总线是否显示可以设定。

④ 梯级边界: 指左总线左边的区域, 其中左列数码为梯级(条)编号, 右列数码为该梯级的首步编号。

⑤ 自动错误检测: 编程时, 在当前选择的梯级左总线处显示一条粗线, 粗线为红色高亮显示, 表示编程出错, 绿色表示输入正确。此外, 梯形图中如果出现错误, 元素的文本也以这种方式来显示。可以通过“工具”菜单“选项”中的“外观”选项卡来定义上述的颜色和显示参数。

⑥ 网格点: 显示各个元素连接处的点。可单击工具栏中的“切换网格”按钮显示网格。

⑦ 选中元素: 单击梯级的一个元素使其高亮, 按住鼠标左键, 可以将其拖动。可以同时选中多个元素, 这些元素可以当做一个块来移动。

在用梯形图编程时, 可以利用工具栏中的触点、线圈、指令等按钮以图形方式输入程序。

在梯形图视图中可进行程序的生成、编辑、监视等。梯形图编程将在 4.3 节中详细介绍。

## (2) 梯形图显示内容和方式的设置

在“工具”菜单中选择“选项”命令, 则显示“选项”对话框, 如图 4.16 所示。可以通过对话框中的 7 个选项卡对梯形图显示内容和方式的参数进行设置。

① “程序”选项卡。在“选项”对话框中选择“程序”选项卡, 则显示如图 4.16 所示对话框。在此对话框中可对梯形图视图进行设置。

选中“显示条和步号”复选框, 将在梯形图左边的梯级边界显示条和步号码。如果不设置, 将显示一个小的梯级边框。

选中“显示条分界线”复选框, 将在每一个梯级的底部显示一条线, 这样, 每一个梯级都有了一个边框。

选中“显示缺省网格”复选框, 将在梯形图的每一个单元格的连接处显示一个点, 其有助于元素的定位。

选中“显示条批注列表”复选框, 将在梯级注释的下方显示一个注释列表(包括梯级里元素的注释和对梯级本身的注释)。这个选项也可以通过工具栏的“显示条批注列表”按钮来快速设置。

选中“允许无窗体的地址引用”复选框, 允许在没有激活“地址引用工具”时使用转移到“下一个引用地址”、“下一个输入”、“下一个输出”、“前一跳转点”命令。如果这一项没有被设置, 在使用这些命令的时候必须激活“地址引用工具”。

选中“水平显示输出指令”复选框, 使特殊指令能够水平显示, 这样, 增加屏幕上显示的梯级数目, 改善程序的可读性, 减少打印所需的纸张数。

“当窗口分割时显示的视图”选项组: 当选择“窗口”菜单中的“分割”命令时, 允许在程序工作区里面显示两个或 4 个视图, 显示的视图由“当窗口分割时显示的视图”选项组中的设置决定。例如, 当在“梯形图编辑器”中选择助记符或符号时, 在“窗口”菜单中选择“分割”命令, 单击梯形图视图时, 则出现两个视图, 即梯形图视图和与其对应的助记符视图或符号表。若在“梯形图编辑器”中选择梯形图, 在“窗口”菜单中选择“分割”命令, 单击梯形图视图, 则出现 4 个梯形图视图。

“右母线”选项组: 选中“显示右母线”复选框, 则显示右总线。通过对初始位置的设置, 可调整梯形图左右总线间的空间。当选中“扩展到最宽的条”复选框时, 右总线的位置将自动匹配本程序段最宽的一个梯级。

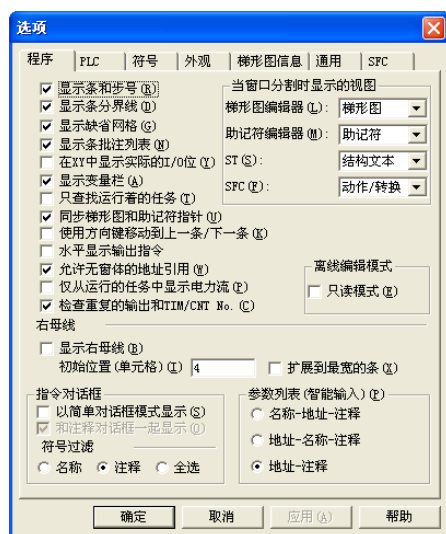


图 4.16 “选项”对话框

② “PLC”选项卡。可设置向工程中添加新的 PLC 时出现的默认的 PLC 类型、CPU 型号及注释等。

③ “符号”选项卡。可设置是否确认所链接的全局符号的修改。

④ 选择“外观”选项卡，显示如图4.17所示的对话框，在对话框中可定义 CX-P 运行环境中的字体和颜色显示。

在“项目”下拉列表框中选择对象，如本地符号、错误、母线等，定义它的“前景色”或“背景色”，颜色也可以按默认设置。

通过“全部复位”按钮，将把所有的颜色设置恢复到系统默认。

通过“梯形图字体”按钮，设置在梯形图程序窗口中显示的字体。

通过“助记符字体”按钮，设置在助记符窗口中显示的字体。

通过“ST 字体”按钮，设置 ST 窗口中显示的字体。

通过“SFC 字体”按钮，设置 SFC 窗口中显示的字体。

通过“单元格宽”滑动条可对梯形图窗口中的单元格宽度进行调整，根据符号名称的典型尺寸来调整单元格在水平方向的大小，通过调整，使文本有一个所需要的显示空间。

⑤ 选择“梯形图信息”选项卡，显示如图4.18所示的对话框，在对话框中可对梯形图中的元素，如触点、线圈、指令和指令操作数的显示信息进行设置。显示的信息越多，梯形图单元格就越大。为了让更多单元格能够被显示，一般只选需要的信息。



图 4.17 “外观”选项卡

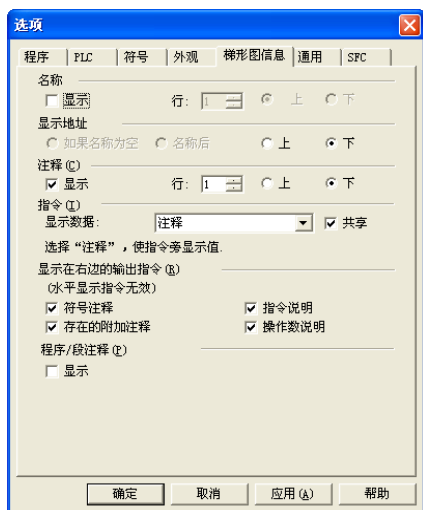


图 4.18 “梯形图信息”选项卡

通过“名称”可以决定显示还是隐藏符号名称，规定显示行数及在元素的上方还是下方显示。

通过“显示地址”可以决定显示还是隐藏名称所对应的存储器，并规定显示的位置。

通过“注释”可以决定显示还是隐藏注释，规定显示的行数及显示的位置。

在监视状态下，通过设置“指令”中的选项来决定指令的监视数据的显示位置。不选“共享”时监视数据显示在名称、地址或注释的下方，选“共享”时监视数据与名称、地址或注释显示在一行。

通过“显示在右边的输出指令”可选择在输出的右边显示的一系列有关输出指令的信息，包括以下选项：“符号注释”、“指令说明”、“存在的附加注释”和“操作数说明”。选中后则显示，否则不显示。

通过“程序/段注释”可决定显示还是隐藏程序/段的注释，选择显示，则在程序的开头处出现注释。

⑥ “通用”选项卡。主要用于改变 CX-P 的窗口环境, 设置 CX-P 创建或打开工程时的视窗风格、显示哪些窗口, 以及最多显示窗口的数目。

⑦ “SFC”选项卡。顺序功能图(Sequential Function Char, SFC)也称流程图或状态转移图, 是一种图形化的功能性说明语言, 专用于描述工业顺序控制程序。OMRON 的 CS、CJ 和 CP1 (除 CP1E 外) 系列 PLC 支持 SFC 编程。

### 3. 助记符视图

助记符视图是一个使用助记符指令进行编程的格式化编辑器。选中工程工作区中的“段 1”, 单击工具栏的“查看助记符”按钮, 显示如图 4.19 所示助记符视图。助记符编程将在 4.3 节中详细介绍。



图 4.19 助记符视图

### 4. 输出窗口

单击查看工具栏上的“切换输出窗口”按钮来激活此窗口, “输出”窗口通常显示在主窗口的下方, 如图 4.20 所示。“输出”窗口下方有“编译”、“寻找报表”和“传送”三个选项, 它们对应三个不同的窗口。再次单击“切换输出窗口”按钮可关闭此窗口。

“编译”窗口: 显示由程序编译产生的输出。选择其中一个错误, 可使梯形图相关部分高亮显示。“编译”窗口也能显示其他信息, 如警告及连接信息。

“寻找报表”窗口: 显示在工程文件内对特定条目进行查找的输出结果。

“传送”窗口: 显示文件或者程序传送的结果。

要清除输出窗口, 可选择快捷菜单中的“清除”命令。

要跳转到“编译”窗口或“寻找报表”窗口中指出的错误源时, 双击窗口中相应的信息, 使用快捷菜单中的“下一个引用”命令, 跳到该窗口下一条信息所指的位置。跳转到的地方在程序工作区中使用高亮来显示。

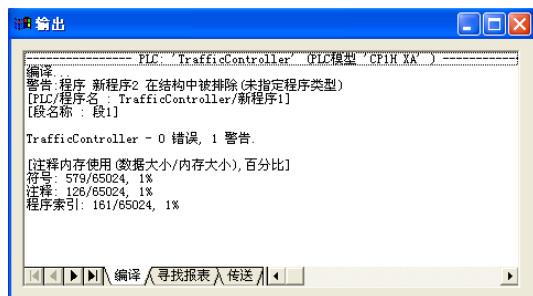


图 4.20 “输出”窗口

### 5. 查看窗口

能够同时监视多个 PLC 中指定的内存区的内容。单击查看工具栏上的“切换查看窗口”按钮来激



活此窗口，如图4.21所示。“查看窗口”窗口通常显示在主窗口的下方，右击“查看窗口”，从其快捷菜单中选择“编辑”命令，在弹出的“编辑”窗口中添加需要查看的内存，可以查看程序执行时该内存的值。再次单击“切换查看窗口”按钮将关闭此窗口。

也可从“视图”菜单的“窗口”子菜单中选择“查看”命令，打开“查看窗口”窗口。

## 6. 地址引用工具

地址引用工具用来显示符号或地址在 PLC 程序中的使用位置。单击查看工具栏上的“显示地址引用工具”按钮来激活此窗口。

用下列步骤来使用地址引用工具。

在梯形图程序里选择一个元素。单击查看工具栏中的“显示地址引用工具”按钮，将显示如图4.22所示“地址引用工具”窗口。该窗口显示出在梯形图程序中所选择的地址的相关信息。



图 4.21 “查看窗口”窗口

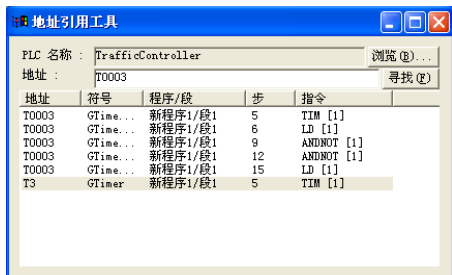


图 4.22 “地址引用工具”窗口

在“地址引用工具”窗口仍然打开时，可以在梯形图程序里面选择另一个元素，“地址引用工具”窗口根据选择会自动刷新。

## 7. 交叉引用表

用来检查内存不同数据区符号的使用。在程序出现问题时，可以用来检查指令设置的值。这可以使编程者能有效地使用存储器资源。

用下列步骤来生成一个“交叉引用表”。

① 单击查看工具栏中的“交叉引用表”按钮来打开如图4.23所示的“交叉引用表”窗口。

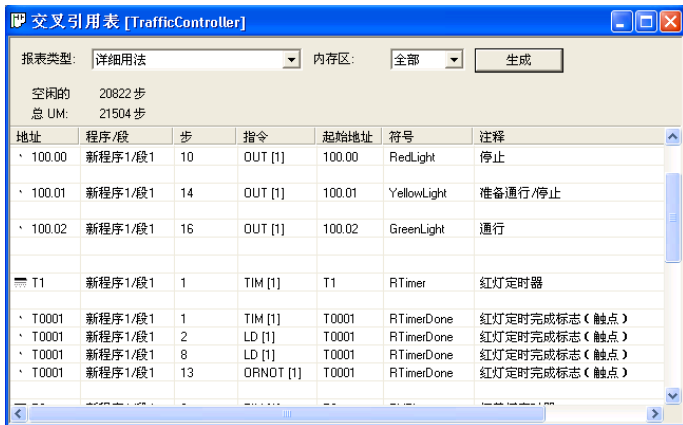


图 4.23 “交叉引用表”窗口

② 在“报表类型”下拉列表框中选择一种使用方法，包括“详细用法”、“用法概况”、“带注释用法”等。

“详细用法”将显示有关 PLC 程序内所用的任一地址的使用信息。将显示程序/段名称、步数、指令类型、操作数起始地址，以及和每一个用法相关的符号。

“用法概况”将显示选定内存区域的总体使用情况，仅仅显示被使用的那一部分内存。对于每一个被使用的内存地址，显示使用数目。符号“ID”表示这个地址已经被分配给一个符号。

“用法(包括未使用过的)概况”将显示内存区域的总体使用情况，包括没有被使用的那一部分内存。

“带注释用法”将显示已使用的内存区域的注释。

“带注释用法(包括未使用过的)”将显示已使用的内存区域的注释，包括没有被使用的那一部分内存。

③ 在“内存区”下拉列表框中选择一个数据区。

④ 单击“生成”按钮来激活交叉引用表。除了地址表以外，在“空闲的”栏还将显示 PLC 内部自由的存储空间容量。PLC 内总的程序存储容量将被显示在“总 UM”栏。

## 4.2.4 其他常用的操作

“转移到”是 CX-P 中常用的一种定位方法。“转移到”适合于梯形图视图和助记符视图，能够将显示转移到程序或程序段中需要的位置。“转移到”为“编辑”菜单下的一个子菜单，此子菜单下有 6 个命令，下面分别介绍。

(1) “条/步号”：可将显示转移到指定的程序或者程序段的条或步，其位置可以通过条(梯级)或者步号码来确定。按以下步骤来进行操作。

① 从“转移到”子菜单中选择“条/步号”命令，显示如图 4.24 所示的“转到”对话框。

② 在对话框中选择梯级的“条”或者“步”号码，也可以通过“浏览(B)...”按钮来选择具有某一注释的梯级。

③ 单击“转到”或“取消”按钮完成转移或放弃操作。

(2) “注释条”：可将显示转移到所选程序段中某一注释所对应的梯级上，按以下步骤来进行操作。

① 从“转移到”的子菜单中选择“注释”命令，则显示如图 4.25 所示的“转移到注释条”对话框。

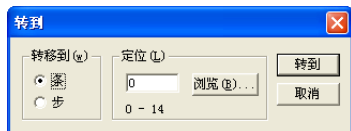


图 4.24 “转到”对话框

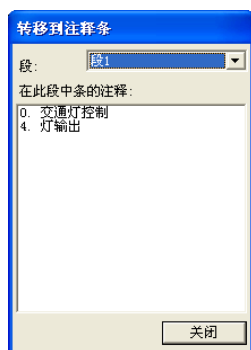


图 4.25 “转移到注释条”对话框

② 在对话框中选择一个程序段时，立即显示出此程序段中所有的梯级注释。

③ 选中某一注释，单击“关闭”按钮可完成转移。

(3) “下一个引用地址”：可将显示转移到当前所选对象的下一个引用。

(4) “下一个输入”：可将显示转移到所选输入对象的下一个引用。如果在当前行以下不存在所选对象，搜索将从顶部开始。



- (5) “下一个输出”：可将显示转移到所选输出对象的下一个引用。如果在当前行以下不存在所选对象，搜索将从顶部开始。
- (6) “前一跳转点”：可将显示转移到前一个跳转点。

4.3 CX-P 编程

用 CX-P 编程时的基本操作有建立一个新工程、生成符号和地址、创建一个梯形图程序、编译程序、将程序传送到 PLC、将 PLC 程序传到计算机、将计算机程序与 PLC 程序比较、在执行程序时进行监视、执行在线编辑(如果需要的话)。

本节以如图4.26所示的交通灯控制为例，介绍 CX-P 的编程。

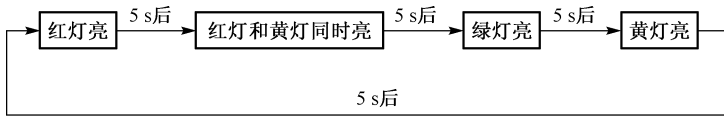


图 4.26 交通灯工作示意图

4.3.1 建立一个新工程


为编写交通灯控制程序，首先建立一个新工程，启动 CX-P 软件，单击“文件”菜单中的“新建”命令或快捷按钮，出现如图4.27所示“变更 PLC”对话框。在此对话框的“设备名称”文本框中输入“TrafficController”，“设备类型”下拉列表框中选择“CP1H”，从其“设定”中选择“XA”；“网络类型”下拉列表框中选择“USB”。单击“确定”按钮，显示如图4.28所示的 CX-P 主窗口，表明建立了一个新工程。



图 4.27 “变更 PLC”对话框

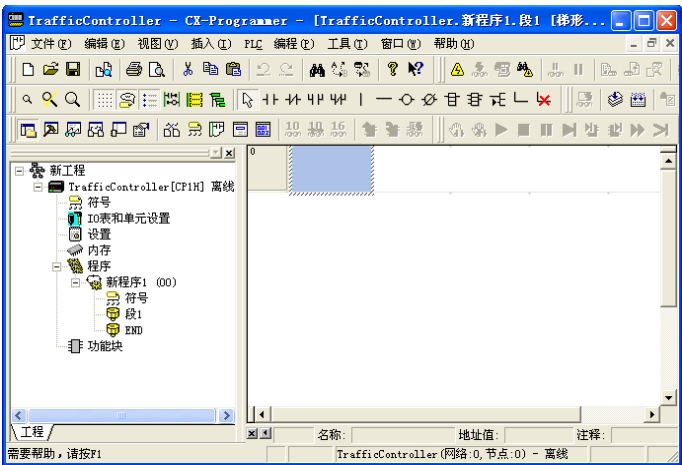


图 4.28 CX-P 主窗口

4.3.2 生成符号和地址

建立一个梯形图程序的重要一步，就是对程序要访问的那些 PLC 数据区进行定义，建立符号与地址、数据的对应关系，输入全局符号表或本地符号表中。

交通灯控制的符号分配见表 4.2，按表将定义的符号输入到本地符号表中。

表 4.2 交通灯控制的符号分配表

符号名称	地址/值	数据类型	注释
RedLight	100.00	BOOL	停止
YellowLight	100.01	BOOL	准备通行/停止
GreenLight	100.02	BOOL	通行
RTimer	001	NUMBER	红灯定时器
RYTimer	002	NUMBER	红黄灯定时器
GTimer	003	NUMBER	绿灯定时器
YTimer	004	NUMBER	黄灯定时器
RTimerDone	TIM001	BOOL	—
RYTimerDone	TIM002	BOOL	—
GTimerDone	TIM003	BOOL	—
YTimerDone	TIM004	BOOL	—
TimeInterval	50	NUMBER	定时时间

双击工程工作区中的“本地符号表”图标，打开本地符号表，单击右键，弹出快捷菜单，选中“插入符号”命令，打开“新符号”对话框，如图4.29所示，根据提示输入即可。交通灯控制的本地符号表如图4.30所示。



图 4.29 “新符号”对话框

名称	数据类型	地址 / 值	机架位置	使用   注释
GreenLight	BOOL	100.02	主机架 : ...	输出   通行
× GTimer	NUMBER	0003		绿灯定时器
· GTimerDone	BOOL	T0003		工作   绿灯定时完成标志
· RedLight	BOOL	100.00	主机架 : ...	输出   停止
× RTimer	NUMBER	0001		红灯定时器
· RTimerDone	BOOL	T0001		工作   红灯定时完成标志
× RYTimer	NUMBER	0002		红黄灯定时器
· RYTimerDone	BOOL	T0002		工作   红黄灯定时完成标志
× TimeInterval	NUMBER	50		定时时间
· YellowLight	BOOL	100.01	主机架 : ...	输出   准备通行/停止
× YTimer	NUMBER	0004		黄灯定时器
· YTimerDone	BOOL	T0004		工作   黄灯定时完成标志

图 4.30 本地符号表

在符号表中除了插入符号，还可以编辑、复制、移动和删除符号。

插入符号的其他方法：

可以选中工程工作区的“全局(或本地)符号表”项目，单击右键，在快捷菜单中选择“插入符号”命令，在弹出的“新符号”对话框中添加新符号。还可以选择“插入”→“符号”命令，在弹出的“新符号”对话框中添加新符号。

4.3.3 程序编辑

1. 梯形图编程

交通灯控制的梯形图程序如图4.31所示。

在工程工作区中双击“段 1”，显示出一个空的梯形图。下面介绍利用如图4.32所示的梯形图工具栏中的按钮来编辑交通灯控制梯形图程序。

(1) 编辑接触点

① 单击梯形图工具栏中的“新常闭接点”按钮，将其放在 0 号梯级的开始位置，将出现如图4.33所示“新的常闭接点”对话框。

② 在符号文本框(第一个空文本框)中输入接触点的符号。可在其下拉列表(表中为全局符号表和本地符号表中已有的符号)中选择符号。本例选择“YTimerDone”。

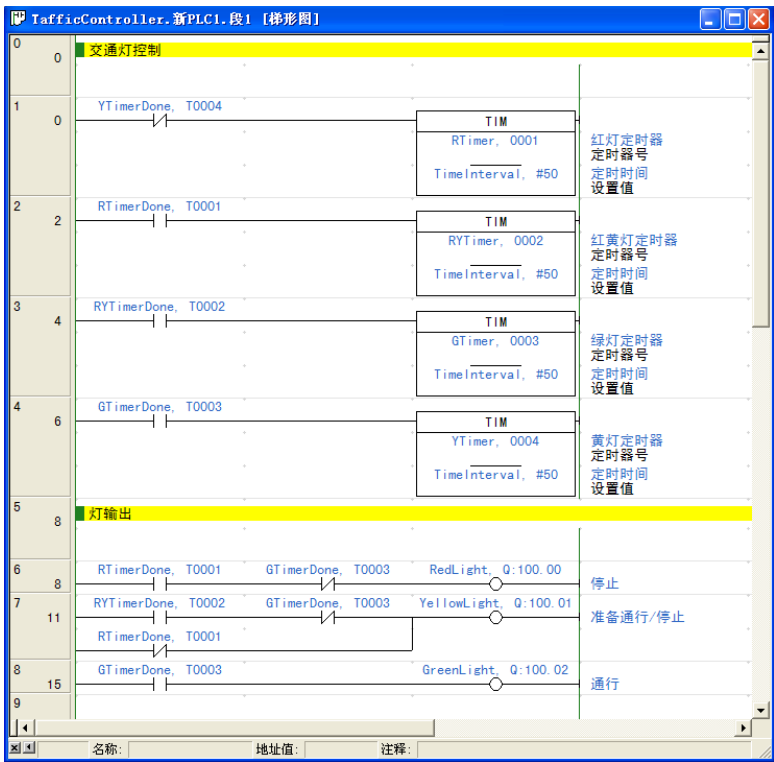


图 4.31 交通灯控制梯形图程序

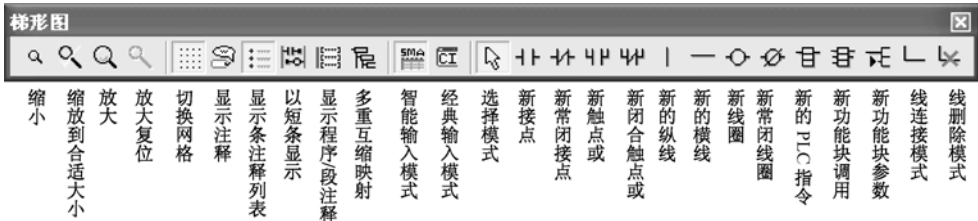


图 4.32 梯形图工具栏

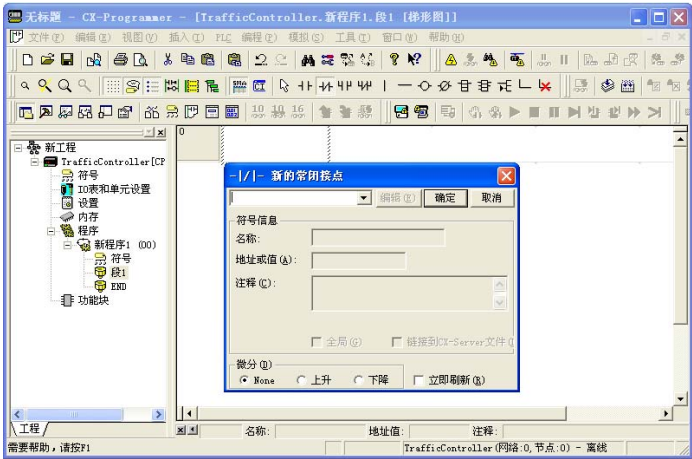


图 4.33 “新的常闭接点”对话框

也可在“符号信息”选项组中,直接输入一个新接触点的符号,再在“地址或值”文本框中输入地址,然后在“注释”编辑框中输入要注释的内容,若要将此符号添加到全局符号表中,则选全局;否则自动添加到本地符号表中。

③ 单击对话框中的“确定”按钮保存操作,单击“取消”按钮则放弃操作。

现在梯级边缘将显示一个红色的记号(颜色可以定义),这是因为该梯级没编辑完,CX-P 认为是一个错误。

可以对已添加接触点进行编辑,双击要编辑的接触点,弹出“编辑接点”对话框。此对话框与“新的常闭接点”对话框的内容完全一样。在此对话框中,可对接触点进行修改。

## (2) 编辑指令


在梯形图工具栏中单击“新的 PLC 指令”按钮,并单击接触点的右侧,则出现如图4.34所示的“新指令”对话框。按以下步骤来输入指令:



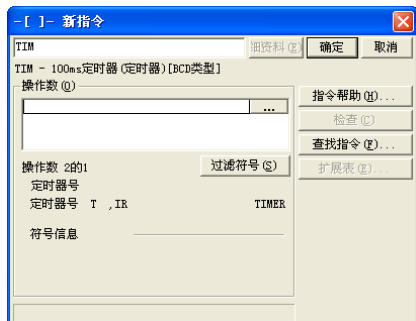
图 4.34 “新指令”对话框

① 在指令文本框中输入指令名称。指令文本框的下方立即出现该指令类型的注释。立即刷新型指令,在指令之前有感叹号“!”;上升沿微分型指令,在指令之前有@符号;下降沿微分型指令,在指令之前有%符号。根据需要可在指令文本框中添加。

也可以单击“查找指令”按钮,“查找指令”对话框被打开,如图4.35所示。“查找指令”对话框的左边是“组”列表框,有 34 类可供选择。选中一组指令,右边“指令”列表框中就出现所选中组的指令。在右边“指令”列表框中选择一条指令后单击“确定”按钮,返回“新指令”对话框,如图4.35(b)所示。



(a)



(b)


图 4.35 “查找指令”对话框

② 见图4.35(b)在“操作数”栏输入指令操作数。操作数可以是符号、地址和数值。

可以单击操作数框右边的“...”按钮来查找符号,将显示一个对话框,能够在这里选择和创建符号。

本例在“新指令”栏输入“TIM”。用“操作数”栏右边的“...”按钮来查找符号,添加已输入的符号“RTimer”和“TimeInterval”,分别作为指令的第1个和第2个操作数。

③ 单击“新指令”对话框中的“确定”按钮完成操作,一条指令就添加到梯形图中了。单击“取消”按钮可放弃操作。

④ 在梯形图工具栏中单击“新的横线”按钮,将接触点和指令连接起来。

此时,在梯级的边缘不再有红色的记号了,这表明该梯级里面已经没有错误了。至此,0号梯级编辑完毕。

前4个梯级都按上述方式进行编辑。

## (3) 编辑线圈

从第5行开始涉及线圈的输入问题,下面介绍线圈的编辑方法。

在 4 号梯级添加一个常开触点“RTimerDone”和一个常闭触点“GTimerDone”后,下一步开始编辑线圈,其步骤为:

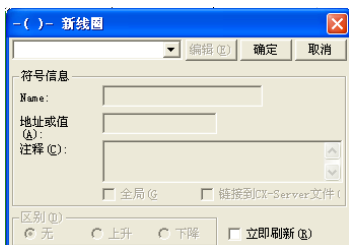




图 4.36 “新线圈”对话框

① 在梯形图工具栏中单击“新线圈”按钮,单击“GTimerDone”的右侧,出现如图4.36所示“新线圈”对话框。

② 在符号下拉列表框中选择线圈的符号。在其下拉列表(表中为全局符号表和本地符号表中已有的符号)中进行选择,本例选择“RedLight”。

在“符号信息”选项组中可以直接输入新符号,这时“地址或值”文本框变为可用状态,在此文本框中输入相应的地址,并将其添加到本地或者全局符号表中。

③ 单击对话框中的“确定”按钮完成编辑线圈的操作,单击“取消”按钮则放弃操作。

在梯形图工具栏中单击“新的横线”按钮,将接触点和线圈连接起来。以下几个梯级可作类似的编辑。

梯形图工具栏中的“选择模式”按钮用来取消所选定的编辑触点、线圈或指令。

#### (4) 给程序添加注释

在编写程序时添加注释,可以提高程序的可读性。选择梯级的属性来给梯级添加注释。选择梯形图元素(触点、线圈和指令)的属性来为其设置注释。文本作为注释,被添加到梯形图中并不被编译。当一个注释被输入,相关元素的右上角将出现一个圆圈。这个圆圈包括一个梯级中标识注释的特定号码。当在“工具”菜单的“选项”命令中做一定设置后,注释内容会出现在圆圈的右部(对输出指令)或者出现在梯级(条)批注列表中。

可以通过梯级快捷菜单中的命令,在所选择梯级的上方或下方插入梯级。

可以通过梯形图元素的快捷菜单中的命令,插入行、插入元素、删除行或删除元素。

## 2. 助记符编程

CX-P 允许在助记符视图中直接输入助记符指令。选中工程工作区中的“段 1”,单击视图工具栏中的“查看助记符”按钮,显示出如图4.37所示助记符视图。助记符编程步骤如下。

① 在“助记符”视图中,把光标定位在相应的位置上。



条	步	指令	操作数	值	注释
0	0	LDNOT	YTimerDone		黄灯定时完成标志(触点)
	1	TIM	0001		红灯定时完成标志(触点)
			#50		
1	2	LD	RTimerDone		红灯定时完成标志(触点)
	3	TIM	0002		红黄灯定时完成标志(触点)
			#50		
2	4	LD	RTimerDone		红黄灯定时完成标志(触点)
	5	TIM	0003		绿灯定时完成标志(触点)
			#50		
3	6	LD	GTimerDone		绿灯定时完成标志(触点)
	7	TIM	0004		黄灯定时完成标志(触点)
			#50		
4	8	LD	RTimerDone		红灯定时完成标志(触点)
	9	ANDNOT	GTimerDone		绿灯定时完成标志(触点)
	10	OUT	RedLight		停止
5	11	LD	RTimerDone		红黄灯定时完成标志(触点)
	12	ANDNOT	GTimerDone		绿灯定时完成标志(触点)
	13	ORNOT	RTimerDone		红灯定时完成标志(触点)
6	14	OUT	YellowLight		准备运行/停止
	15	LD	GTimerDone		绿灯定时完成标志(触点)
	16	OUT	GreenLight		运行
8	17	END	(001)		

图 4.37 CX-P 助记符编程

- ② 按 Enter 键，即进入编辑模式。
  - ③ 编辑或者输入新的指令。一个助记符指令由一个指令名称和用空格分隔开来的操作数组成，如 LD RTimerDone。
  - ④ 再次按 Enter 键，把光标移动到下一行或者使用键盘上的“↑”或者“↓”键把光标移动到另一行，所做的输入被保存。
  - ⑤ 当输入完成以后，按 Esc 键来结束编辑模式。
- 对助记符视图，在梯级的开始输入梯级注释时，先输入字符“'”后输入文本；要给一个元素输入注释，输入字符“//”，然后输入文本。

3. 梯级的语句列表编程

CX-P 支持以语句列表的方式来输入梯级，即块程序。梯级的语句列表视图可以用来代替助记符视图，但是其不支持程序监视。可按照以下步骤以语句列表格式来编辑梯级。

- ① 选择一个梯级，从其快捷菜单中选择“显示条按照”→“说明列表”命令，梯级将以语句列表的方式显示。将光标移动到相应的行，按 Enter 键就可以编辑指令。使用方向键来移动光标，修改文本。
- ② 语句列表中的项目被不断编译，有可能显示错误标记。
- ③ 按 Esc 键退出编辑模式，完成编辑。
- ④ 也可以重新将梯级显示为梯形图模式。从其快捷菜单中选择“显示条按照”→“梯形图”命令，重新切换到梯形图格式。

4.3.4 程序编译

如图4.38所示，程序工具栏上有两个编译按钮：“编译程序”和“编译 PLC 程序”。前者只是编译 PLC 下的单个程序，后者则编译 PLC 所有的程序。单击这两个按钮中的一个，编译结果显示在输出窗口的“编译”窗口中。

程序编译时，通过选择“PLC”→“程序检查选项”命令，显示“程序检查选项”对话框，如图 4.39 所示。可在检查级“A”、“B”、“C”（“A”最多，“B”次之，“C”最少）或“定制”之间选择。当选择“定制”时，可任意选择检查项。编译时将按选定的项目检查程序的正确性。

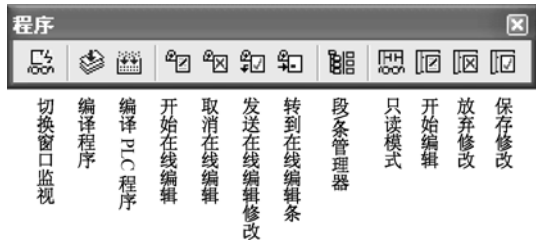


图 4.38 程序工具栏

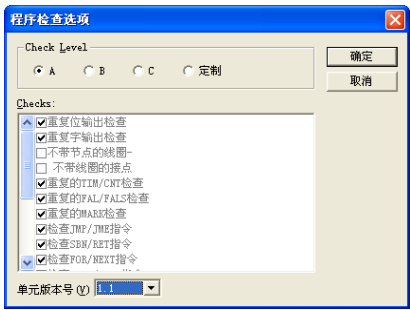


图 4.39 “程序检查选项”对话框

4.3.5 在线工作

如图 4.40 所示的 PLC 工具栏，是用于 CX-P 在线工作的。

1. 离线方式与在线方式

离线方式下，CX-P 不与 PLC 进行通信；在线方式下，CX-P 与 PLC 进行通信。究竟选用何种方



式根据需要而定,例如,修改符号表,必须在离线方式下进行;而要监控程序运行,则应在在线方式下进行。

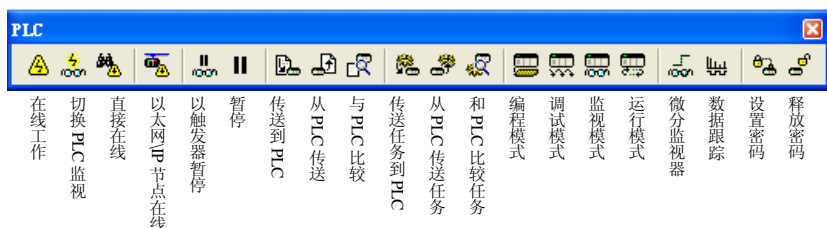


图 4.40 PLC 工具栏

在工程工作区选中“PLC”后,单击图4.40 PLC 工具栏中的“在线工作”按钮,将出现一个确认对话框,单击“是”按钮,则计算机与 PLC 联机通信,处于在线方式;再单击“在线工作”按钮,则转换到离线方式。

## 2. PLC 操作模式

PLC 能够被设置成下列 4 种工作模式中的一种。

- ① 编程模式。这种模式下,PLC 不执行程序,可下载程序和数据。CX-P 可向 PLC 下载程序、进行 PLC 设定和配置 I/O 表等。
- ② 调试模式。这种模式对 CV 系列 PLC 可用,能够实现用户程序的基本调试。
- ③ 监视模式。这种模式下,可对运行的程序进行监视,在线编辑必须在这种模式下进行。
- ④ 运行模式。这种模式下,PLC 执行用户程序。对运行的程序只能监视,不能改写 PLC 内部的数据。PLC 的 4 种工作模式可通过单击 PLC 工具栏中的相应按钮来切换。

## 3. 把程序传送到 PLC(下载)

- ① 选中工程工作区里的“PLC”。
- ② 单击 PLC 工具栏中的“在线工作”按钮,与 PLC 进行连接。将出现一个确认对话框,单击“是”按钮。由于在线时一般不允许编辑,所以程序变成灰色。
- ③ 单击 PLC 工具栏上的“编程模式”按钮,把 PLC 的操作模式设为编程。如果未做这一步,那么 CX-P 将自动把 PLC 设置成此模式。
- ④ 单击 PLC 工具栏上的“传送到 PLC”按钮,将显示“下载选项”对话框,可以选择的项目有程序、设置、特殊单元设置、符号、注释等。
- ⑤ 按照需要选择后,单击“确定”按钮,出现“下载”窗口。
- ⑥ 当下载成功后,单击“确定”按钮,结束下载。

## 4. 从 PLC 传送程序到计算机(上载)

- ① 选中工程工作区里的“PLC”。
- ② 单击 PLC 工具栏中的“在线工作”按钮,与 PLC 进行连接。将出现一个确认对话框,单击“是”按钮。
- ③ 单击 PLC 工具栏里的“从 PLC 传送”按钮,将显示“上载选项”对话框,可以选择的项目有程序、设置、I/O 表、特殊单元设置、符号、注释等。
- ④ 按照需要选择后,单击“确定”按钮确认操作,出现确认传送对话框。
- ⑤ 单击“确定”按钮确认操作,出现“上载”窗口。
- ⑥ 当上载成功后,单击“确定”按钮,结束上载。

## 5. 与 PLC 比较程序

① 选中工程工作区里的“PLC”。

② 单击 PLC 工具栏中“与 PLC 比较”按钮，将显示“比较选项”对话框，可以选择的项目是程序中的各个任务和每个程序中不同程序段。

③ 按照需要选择后，单击“确定”按钮确认操作。如果出现“比较失败”确认对话框，则上位机程序与 PLC 中程序之间的程序不同。单击“确认”按钮，比较细节将显示在弹出的“比较结果”窗口中。

## 6. 在线编辑

在线状态下，程序工作区变成灰色，一般不能被直接编辑。对少量改动(仅限 1 个梯级范围)，可以选择在线编辑特性来修改梯形图程序。

当使用在线编辑功能时，要使 PLC 运行在编程或监视模式下，而不能在运行模式下。使用以下步骤进行在线编辑。

① 拖动鼠标，选择要编辑的梯级。

② 单击 PLC 工具栏中“与 PLC 比较”按钮，以确认编辑区域的内容和 PLC 内的相同。

③ 单击程序工具栏中“转到在线编辑条”按钮，梯级的背景将改变，表明其现在已经是一个可编辑区，此时可以对梯级进行编辑。此区域以外的梯级不能改变，但是可以把这些梯级里面的元素复制到可编辑梯级中去。

④ 当对编辑结果满意时，单击程序工具栏中“发送在线编辑修改”按钮，所编辑的内容将被检查并且被传送到 PLC，一旦这些改变被传送到 PLC，编辑区域再次变成灰色。

若想取消所做的编辑，单击程序工具栏中的“取消在线编辑”按钮，可以取消在确定改变之前所做的任何在线编辑，编辑区域也将变成灰色。在线编辑不能改变符号的地址和类型。

## 7. 程序监视

一旦程序运行，就可以对其进行监视。可按以下步骤启动和停止程序监视。

① 在工程工作区中双击某一程序段，在程序工作区显示梯形图程序。

② 单击 PLC 工具栏中的“在线工作”按钮，与 PLC 进行连接，将出现一个确定对话框，单击“是”按钮。

③ 单击 PLC 工具栏中的“监视模式”或“运行模式”按钮，只能在这两种模式下进行程序监视。

④ 单击 PLC 工具栏中的“切换 PLC 监视”按钮，可监视梯形图中数据的变化和程序的执行过程，再次单击“切换 PLC 监视”按钮停止监视。

## 8. 暂停监视

暂停监视能够将普通监视及时冻结在某一点，在检查程序的逻辑时很有用处。可以通过手动或者触发条件来触发暂停监视功能，下面介绍暂停监视操作。

确认打开“梯形图”程序，并处在监视模式下。

① 选择一定的梯级范围以便于监视。

② 单击 PLC 工具栏中的“以触发器暂停”按钮，出现“暂停监视设置”对话框，选择触发类型：手动或者触发器。

手动：选择“手动”，单击“确定”按钮后，开始监视。等到屏幕上出现感兴趣内容时，单击 PLC 工具栏中的“暂停”按钮，暂停功能发生作用。要恢复监视，可再次单击“暂停”按钮，监视将被恢复，等待另一次触发暂停监视。

触发器：在“地址和姓名”文本框中输入一个地址，或者使用浏览器来定位一个符号。选择“条



件”类型：“上升沿”、“下降沿”或输入触发的“值”。当暂停监视功能工作时，监视仅仅发生在所选区域，选择区域以外的地方无效。要恢复完全监视，可再次单击“以触发器暂停”按钮。

当使用触发器类型时，也可以通过单击 PLC 工具栏中的“暂停”按钮来手动暂停。

4.4 CX-P 的功能块编程

功能块是一个包含标准处理功能的基本单元，该标准处理功能事先已定义好。用户可以将已定义好的功能块嵌入到梯形图程序中调用，同时设置执行功能块的输入/输出条件。功能块不包含实际地址，只有变量。用户可以在变量中设置地址或常数。

采用 CX-P 将单个功能块保存为单个文件，保存后的功能块可用于其他 PLC 程序中。因此，可将不同的功能块集成成一个功能块库。编程需要时，调用即可。

功能块编程的内容为：创建功能块和调用功能块。而创建一个功能块需要进行变量定义和算法设计，如图4.41所示。

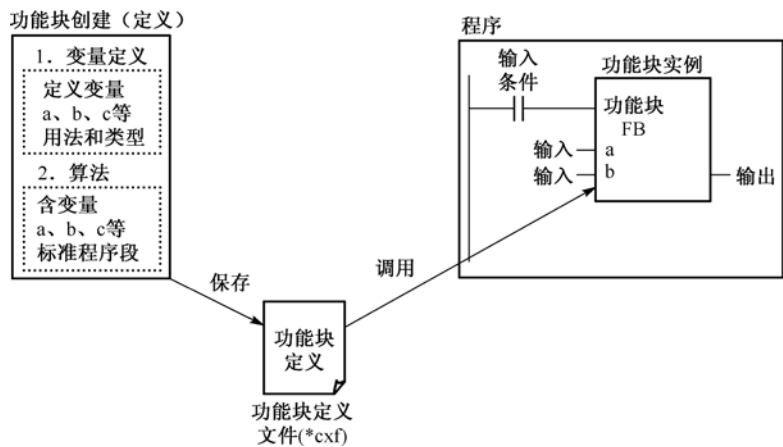


图 4.41 功能块编程示意图

4.4.1 创建功能块

每个功能块的创建(定义)都包含了变量定义和算法，如图4.42所示。

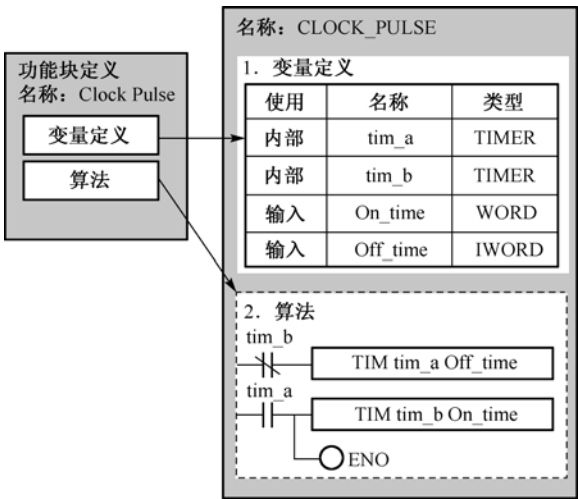


图 4.42 Clock\_Pulse 功能块的创建示意图

下面以“时钟脉冲(Clock\_Pulse)”为例，介绍功能块的创建过程。

1. 创建新功能块

① 进入 CX-P 主窗口，如图4.43所示，注意工程工作区的最后一个项目为“功能块”。

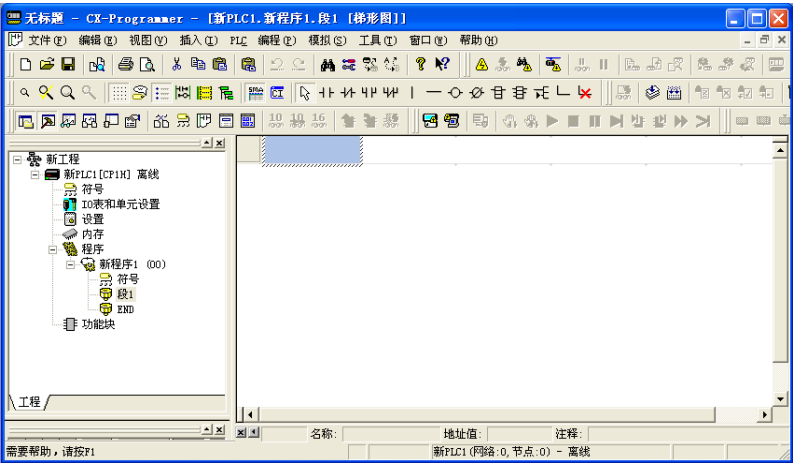


图 4.43 CX-P 主窗口

② 选中工程工作区“功能块”图标，单击鼠标右键，在弹出菜单中选择“插入功能块”→“梯形图”命令，操作如图4.44所示，成功插入功能块 1，如图4.45所示。

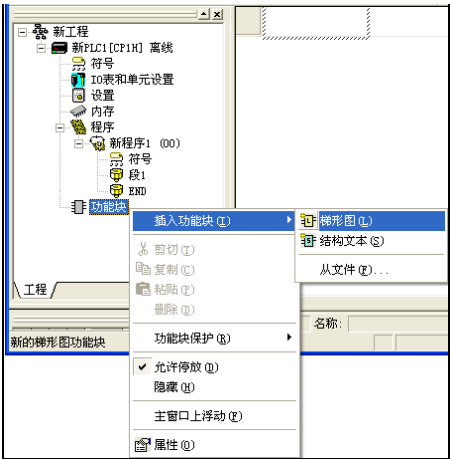


图 4.44 插入功能块操作

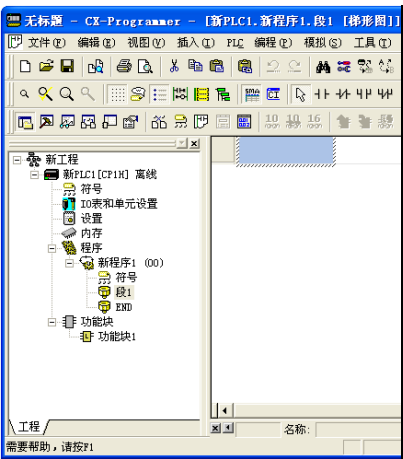
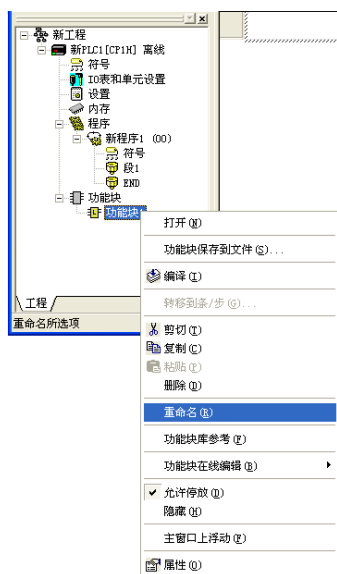


图 4.45 成功插入功能块 1

③ 命名功能块。成功插入的“功能块 1”是默认的功能块名。选中“功能块 1”，单击鼠标右键，在弹出菜单中选择“重命名”命令，如图4.46(a)所示，得到图4.46(b)，输入“时钟脉冲”，更改结果如图4.47所示。

④ 上述②和③是创建新功能块的步骤，但是也可以直接调用功能块库中已经建立好的文件(格式为\*.cxf)。步骤如下：选中工程工作区的“功能块”图标，单击鼠标右键，在弹出菜单中选择“插入功能块”→“从文件”命令，将弹出如图4.48所示的对话框，从中选择需要的功能块文件，例如选择 omronlib → PLC → CLK → “\_CLK004\_CheckNode6210.cxf”文件，如图 4.49 所示，成功插入 \_CLK004\_CheckNode6210.cxf 功能块。



(a)

图 4.46 功能块命名操作



(b)



图 4.47 功能块命名完成

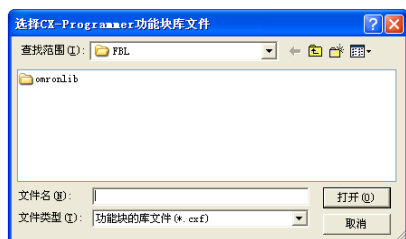


图 4.48 “选择 CX-Programmer 功能块库文件”对话框

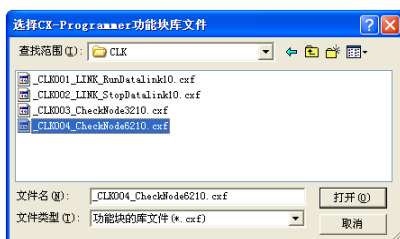


图 4.49 选择功能块库文件

使用功能块库文件时，双击工程工作区中插入的功能块\_CLK004\_CheckNode6210.cxf，在程序工作区中显示上方的变量表窗口和下方的梯形图编辑区窗口。默认状态下不显示功能块库文件内部。若需要显示时，则右键单击工程工作区中的该功能块文件图标，在弹出的菜单中选择“属性”命令。并在属性窗口中选“显示功能块的内部”复选框。则程序工作区呈现灰色，表示不能进行编辑操作，如图4.50所示。

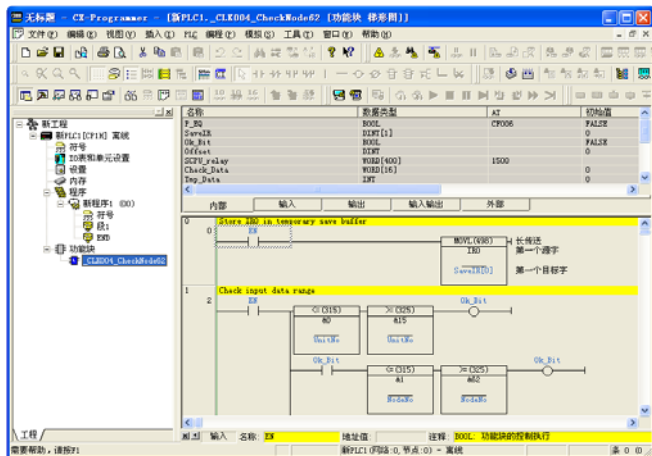


图 4.50 库文件中的功能块定义

2. 定义变量

(1) 变量类型

变量分为如下 4 种类型。

- ① 内部变量：只能在功能块内部使用的变量，不能直接用来传输数据至 I/O 参数或从 I/O 参数中上传数据。
- ② 输入变量：可以从实例外的输入参数中输入数据。每个功能块都有一个默认输入变量 EN(Enable)，EN 为使能变量，功能块创建时自动生成，不需要用户定义，当 EN 处于 ON 时，功能块执行。
- ③ 输出变量：可以输出数据至实例外的输出参数。每个功能块都有一个默认输出变量 ENO(Enable Out)，ENO 为使能输出变量，功能块创建时自动生成，不需要用户定义，功能块执行时，ENO 处于 ON 状态。
- ④ 外部变量：事先由 CX-P 寄存的系统定义变量，如状态标志或用户定义的全局符号。

选中上面完成命名的“时钟脉冲”图标，双击，得到功能块定义窗口，如图4.51 所示。它由变量表和梯形图编辑区两个部分组成。变量定义采用变量表的形式在变量表中写入，算法设计采用梯形图编程方式写在梯形图编辑区内。

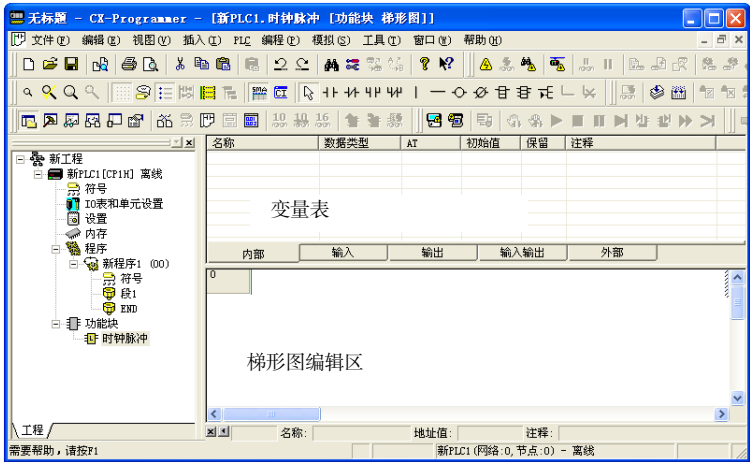


图 4.51 功能块定义窗口

(2) 变量定义

变量表底部有 5 个选项：“内部”、“输入”、“输出”、“输入输出”和“外部”，单击“内部”选项，得到内部变量表。在此可以定义内部变量。将光标置于内部变量表格中，单击鼠标右键，弹出如图4.52 所示菜单，选择“插入变量”命令，弹出“新变量”对话框，如图4.53 所示。

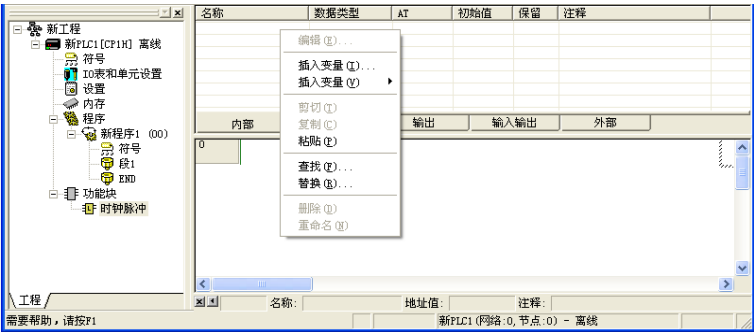


图 4.52 添加内部变量操作

该窗口包含名称、数据类型、使用、初始值、保留、注释和高级等选项。

在“名称”文本框中输入定时器 1 的变量名“tim\_a”，在“数据类型”下拉列表框中选择定时器 1 为“TIMER”，如图 4.54 所示。数据类型的种类及属性见表 4.3。

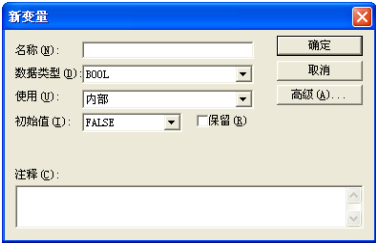


图 4.53 “新变量”对话框

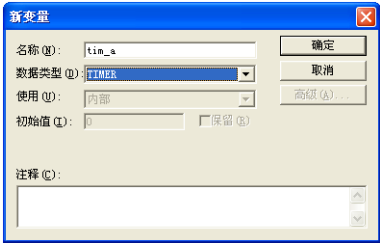


图 4.54 设置新变量

表 4.3 变量数据类型

数据类型	内容	大小	输入	输出	内部
BOOL	位数据	1 位	可以	可以	可以
INT	整数	16 位	可以	可以	可以
UINT	无符号整数	16 位	可以	可以	可以
DINT	双整数	32 位	可以	可以	可以
UDINT	无符号双整数	32 位	可以	可以	可以
LINT	长(4 个字)整数	64 位	可以	可以	可以
ULINT	无符号长整数	64 位	可以	可以	可以
WORD	16 位整数	16 位	可以	可以	可以
DWORD	32 位整数	32 位	可以	可以	可以
LWORD	64 位整数	64 位	可以	可以	可以
REAL	实数	32 位	可以	可以	可以
LREAL	长实数	64 位	可以	可以	可以
TIMER	定时器	标记: 1 位 PV: 16 位	不支持	不支持	可以
COUNTER	计数器	标记: 1 位 PV: 16 位	不支持	不支持	可以

图 4.54 中定时器 1 的“使用”下拉列表框中显示为“内部”，代表该变量类型为内部变量，由表 4.3 也可以查到数据类型“TIMER”只支持内部变量。

“初始值”是指在初次执行功能块时设置的变量值。在功能块运行过程中可以修改。该项为灰色时，表示不能对其进行编辑。

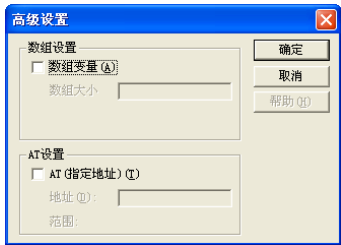


图 4.55 “高级设置”对话框

“保留”是指当 PLC 重新加电运行时，若要某些变量的数据保持不变，则选择此项。该项为灰色时，表示不能对其进行编辑。

“注释”用于介绍变量用途，可以选用。

单击“高级”按钮，弹出“高级设置”对话框，如图 4.55 所示。在此可以设置“数组变量”和“AT(指定地址)”。

单击“高级设置”对话框的“确定”按钮，回到图 4.54 的“新变量”对话框，再单击“新变量”对话框的“确定”按钮，名称为“tim\_a”的变量就注册到了内部变量表中，如图 4.56 所示。



跳转指令：JMP、CJP、CJPN 和 JME。

步进指令：STEP 和 SNXT。

立即刷新指令：（！）。

I/O REFRESH 指令：IQRF。

TMHH 和 TMHHX 指令：


CV 地址转换指令：FRMCV 和 TOCV。

堆栈记录位置指令：PUSH、FIFO、LIFO、SETR 和 GETR。

FAILURE POINT DETECTION 指令：FPD 等。

变址寄存器设定指令：MOVRW。

按以下步骤在功能块中创建时钟脉冲算法。

在图 4.56 中，用鼠标单击梯形图编辑区，此时梯形图工具栏由灰变黑，用鼠标单击选中梯形图工具栏的常闭触点图标，然后，单击第 0 条(梯级)最左边的格子，放置一个常闭触点，弹出一个“新的常闭触点”对话框，如图 4.59 所示。在此对话框的下拉列表框中选择已注册变量“tim\_b”，如图 4.60 所示，单击“确定”按钮后，常闭触点“tim\_b”已经输入到第一行首部，如图 4.61 所示，并且梯级的边缘出现红色竖线，表明梯级程序没写完整。

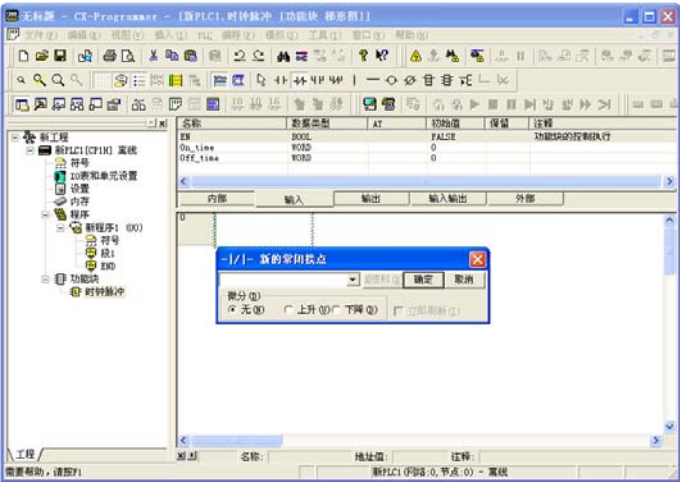


图 4.59 “新的常闭触点”对话框

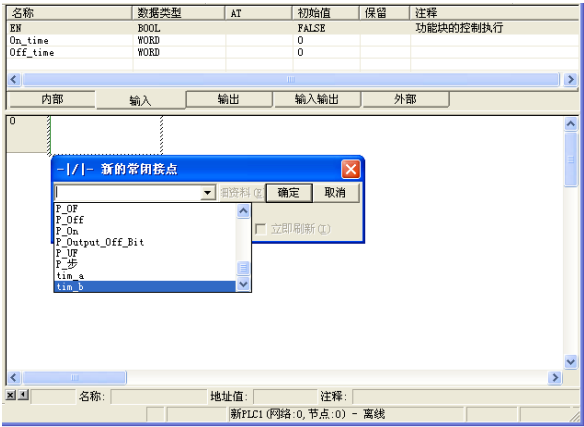



图 4.60 选择变量



图 4.61 输入的常闭触点“tim\_b”

用鼠标单击选中梯形图工具栏的“新的 PLC 指令”按钮, 单击常闭触点“tim\_b”的右边位置, 放置指令, 则出现如图4.62所示的“新指令”对话框。按以下步骤来输入指令。

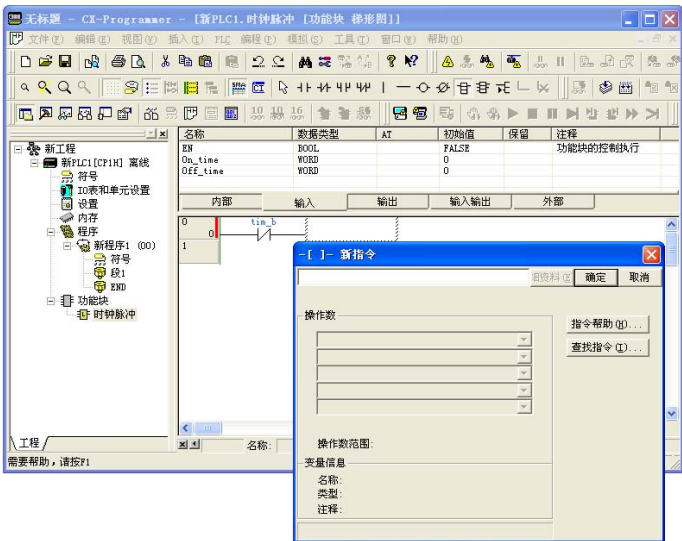


图 4.62 “新指令”对话框 1

① 在图4.62 指令文本框(第一个空文本框)输入指令名称“TIM”。  
也可以单击“查找指令”按钮, 弹出“查找指令”对话框, 如图4.63所示。“查找指令”对话框的左边是指令组, 有 34 类可供选择。选中左边“定时器/计数器”, 再在右边选择“TIM”, 然后单击“确定”按钮, 返回“新指令”对话框, 如图4.64所示。

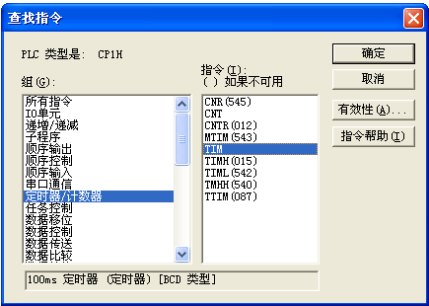


图 4.63 “查找指令”对话框




图 4.64 “新指令”对话框 2





图 4.65 输入定时器操作数

② 在“操作数”选项组的第 1 行输入第 1 个指令操作数“tim\_a”，也可以从第 1 行的下拉列表中选择“tim\_a”；在第 2 行输入第 2 个指令操作数“Off\_time”，也可以从第 2 行的下拉列表中选择“Off\_time”，如图 4.65 所示。单击“确认”按钮完成操作。

此时，在梯级的边缘不再有红色的竖线，这表明该梯级里面已经没有错误了。至此，0 号梯级编辑完毕，如图 4.66 所示，一条指令就添加到梯形图中了(如需要，在梯形图工具栏中选择“新的横线”按钮)，将接触点和指令连接起来)。

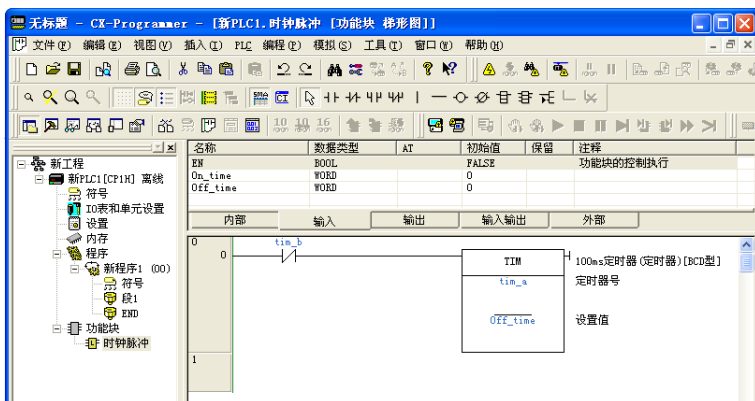


图 4.66 显示 0 号梯级程序

③ 按照以上步骤完成“时钟脉冲”的算法设计，如图 4.67 所示。

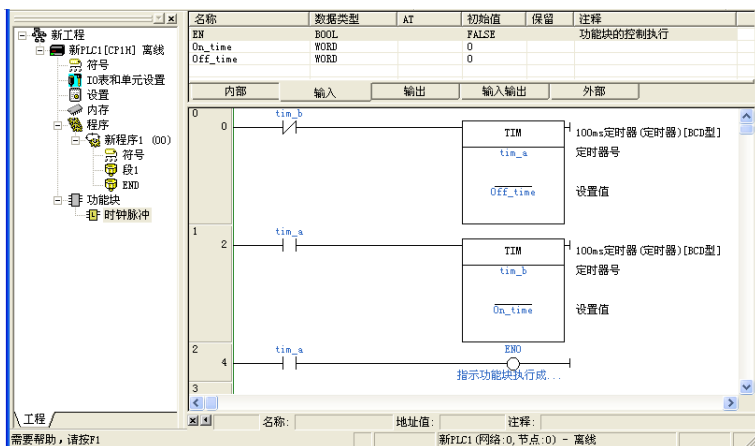


图 4.67 时钟脉冲梯形图程序

#### 4. 功能块文件的保存

利用 CX-P 可以将创建好的功能块保存成扩展名为\*.cxf 的库文件。保存好的功能块库文件可以被其他应用程序调用。其操作步骤如下。

在工程工作区中右击已创建好的功能块，在弹出的菜单中选择“功能块保存到文件”命令，如图 4.68 所示。找到 OMRON 库文件“FBL”，“选择 CX-Programmer 功能块库文件”对话框如图 4.69 所示，在图 4.69 中输入文件名“时钟脉冲”，单击“保存”按钮，完成保存操作。

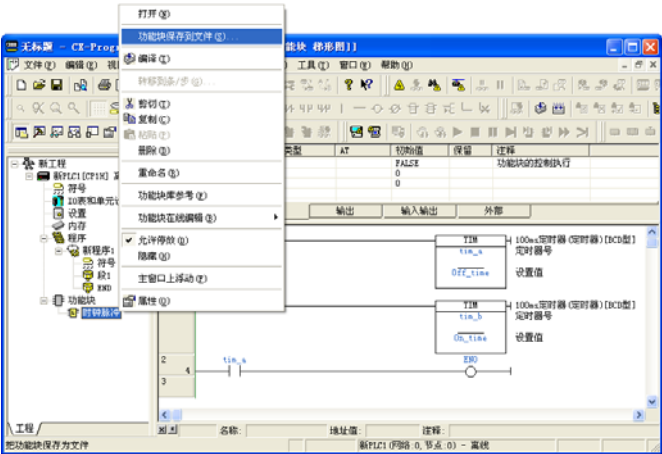


图 4.68 保存“功能块”操作

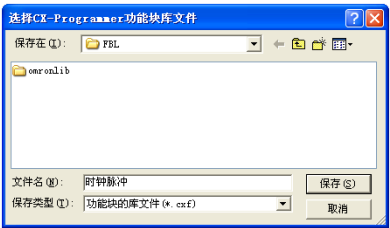


图 4.69 “选择 CX-Programmer 功能块库文件”对话框

4.4.2 功能块的使用

1. 功能块文件的调用

要将实际功能块应用在程序中，必须创建一个功能块图并将其插入程序中，嵌入程序中的每个功能块定义称为“实例”或“功能块实例”。每个实例有一个标识符，称为“实例名”。

在调用了功能块的梯形图中，应在程序结构上注意以下两点：

- ① 功能块左侧不允许有程序分支，仅允许在右侧有分支，如图4.70所示。
- ② 在梯形图的某一逻辑行中只允许调用一个功能块，如图4.71所示。

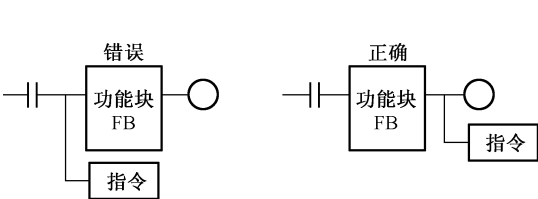


图 4.70 功能块分支结构示意图

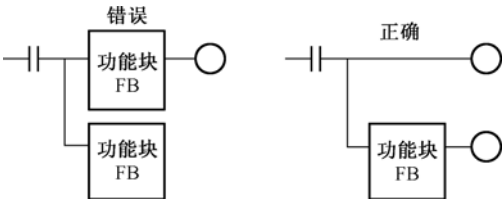


图 4.71 功能块数目示意图

以时钟脉冲为例介绍功能块的调用方法，目的是产生可变占空比的脉冲，操作步骤如下。

- ① 在工程工作区中，双击“段 1”，如图4.72所示。添加地址为 0.00 的常开触点，如图4.73所示。

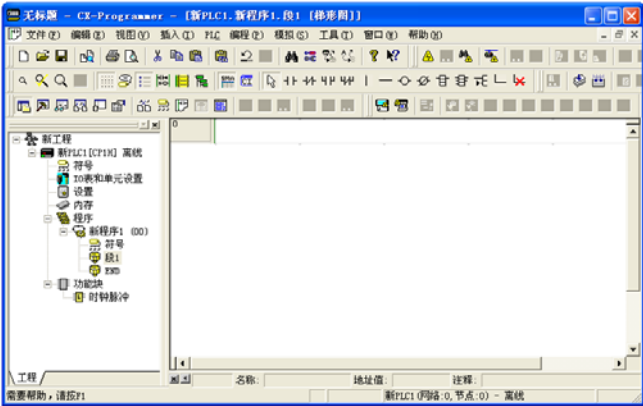


图 4.72 梯形图编辑窗口

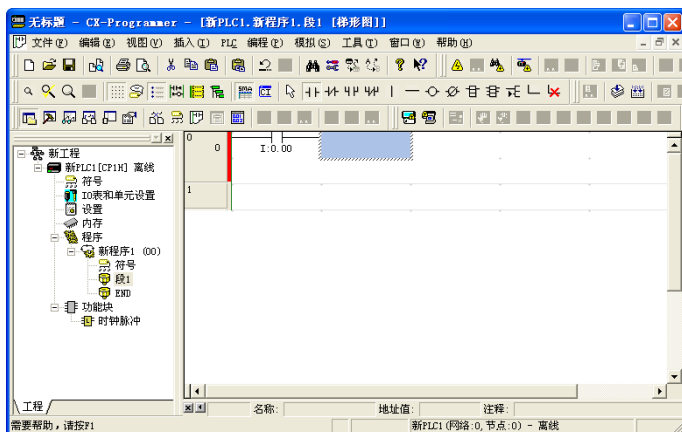


图 4.73 添加常开触点 0.00

② 在图4.73光标处插入功能块时，须单击“插入”菜单的“功能块调用”命令，如图4.74所示。则弹出“新功能块调用”对话框，如图4.75所示。

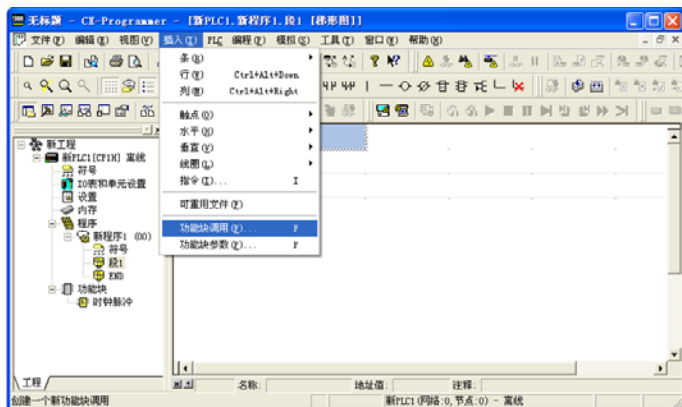


图 4.74 调用功能块操作



图 4.75 “新功能块调用”对话框

③ 在“新功能块调用”对话框的“功能块实例”文本框中输入实例名“CP”，单击“确定”按钮，新功能块调用完成，如图4.76所示。

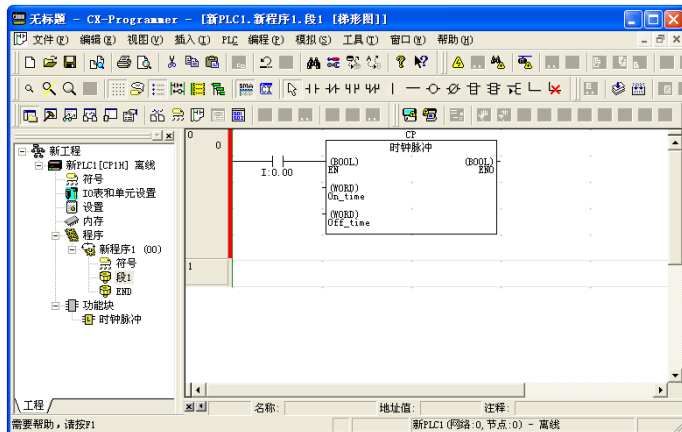


图 4.76 功能块实例调用完成

④ 设置功能块参数，创建功能块实例后，要设置输入变量和输出变量的参数以激活外部 I/O，具体操作如下：

- 输入变量位于实例的左侧而输出变量位于实例右侧。将光标置于待设置的参数处，按 Enter 键，将弹出如图4.77所示的“新参数”对话框。

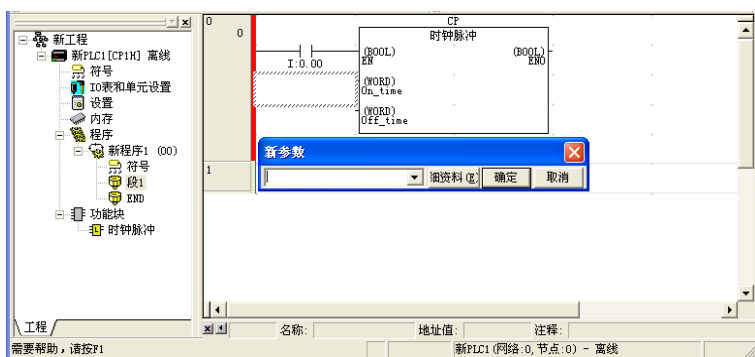


图 4.77 “新参数”对话框

- 设置定时器 tim\_a 的“On\_time”时间为 5 s，输入“#50”，单击“确定”按钮，设置完成，如图4.78所示。

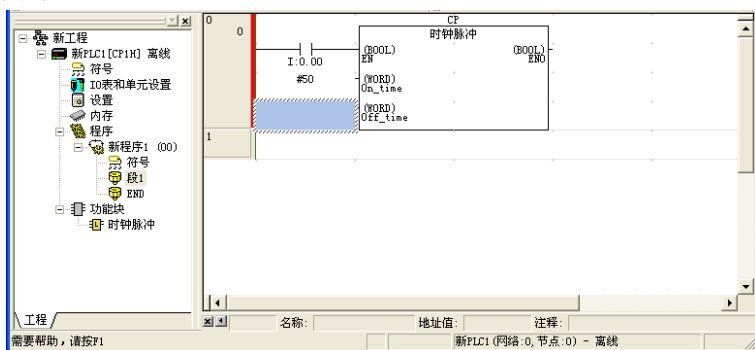


图 4.78 “On\_time”参数设置结果

- 按上一步的方法，设置定时器 tim\_a 的“Off\_time”时间为 2 s，结果如图4.79所示。

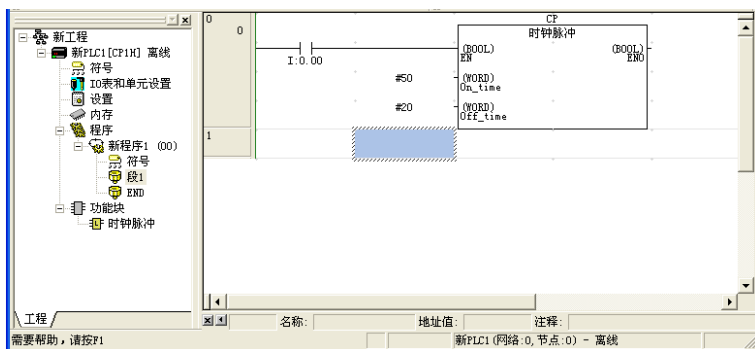


图 4.79 输入参数设置结果

- 单击梯形图工具栏中的“新线圈”按钮，在功能块的输出变量后加入地址为 100.00 的输出线圈，结果如图4.80所示。

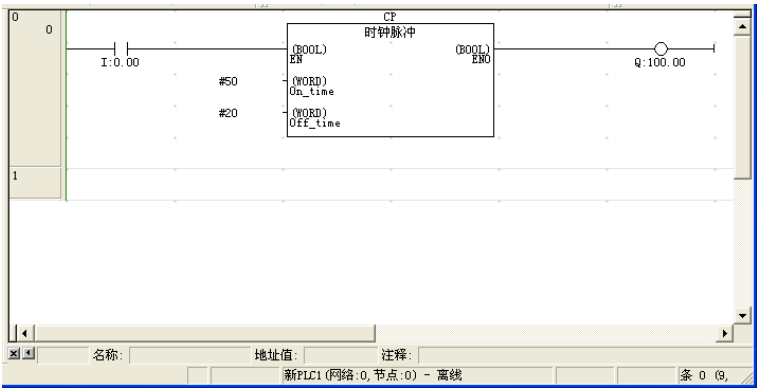


图 4.80 功能块参数设置结果

- 按上述步骤利用时钟脉冲功能块实现可调占空比的控制程序，梯形图如图4.81所示。

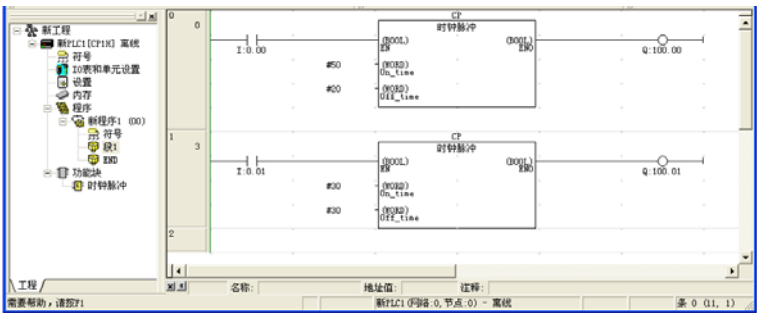


图 4.81 可调占空比的脉冲产生程序

- 编译功能块，在工程窗口中右键单击“时钟脉冲”功能块，在弹出的菜单中选择“编译”命令，开始编译功能块，操作如图4.82所示。结果显示在输出窗口的编译页面中，如图4.83所示。

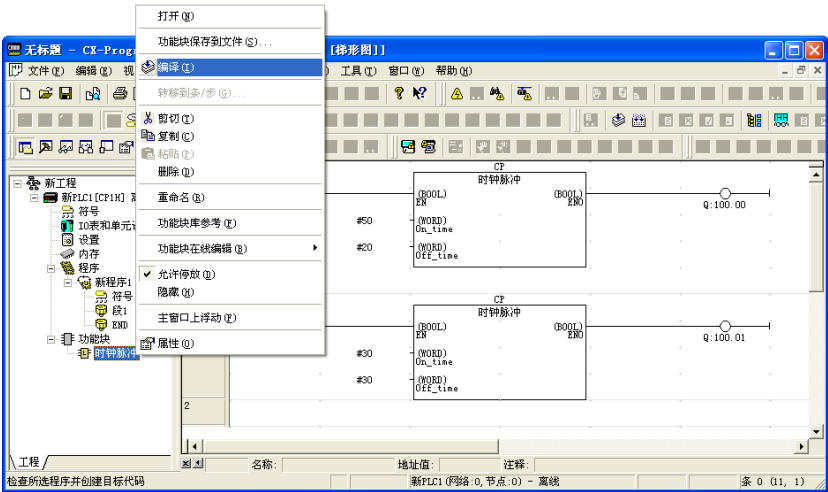


图 4.82 编译“功能块”操作

## 2. 功能块文件的在线监视

在线监视该程序的步骤如下。


① 首先单击“在线工作”按钮，在弹出的对话框中单击“是”按钮，将 CX-P 与 PLC 连接。单击 PLC 工具栏上的“传送到 PLC”按钮，将显示“下载选项”对话框，选择下载项目后，单击“确定”按钮，开始下载，完成后，单击“确定”按钮，界面如图4.84所示。



图 4.83 输出窗口的“功能块”编译页面

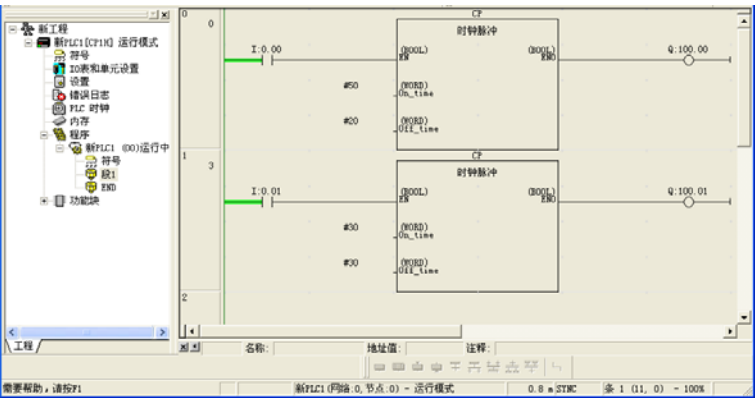


图 4.84 在线工作界面

② 在常开触点 0.00 处，单击鼠标右键，将弹出快捷菜单。选择“强制”→“ON”命令，操作如图4.85所示，结果如图4.86所示。



图 4.85 强制触点为 ON 操作

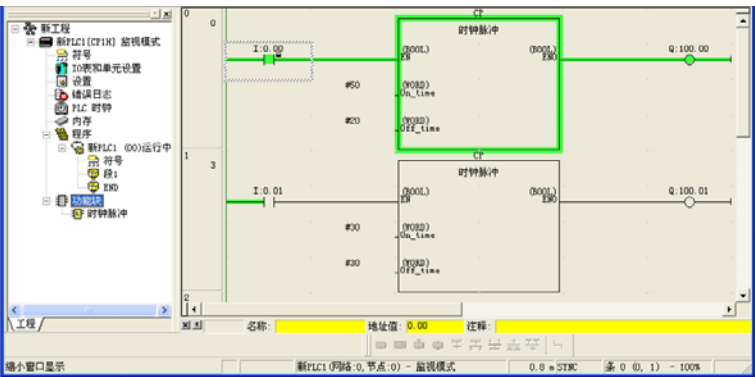


图 4.86 功能块在线运行

③ 双击实例，进入功能块内部进行在线监视，如图4.87所示。

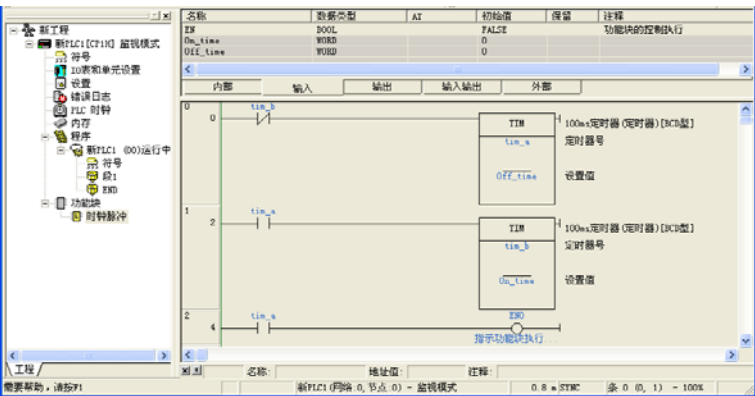


图 4.87 功能块内部在线监视

④ 为了便于监视，从“视图”菜单中选择“窗口”→“功能块/结构实例浏览器”命令，如图4.88所示。执行结果如图4.89所示。

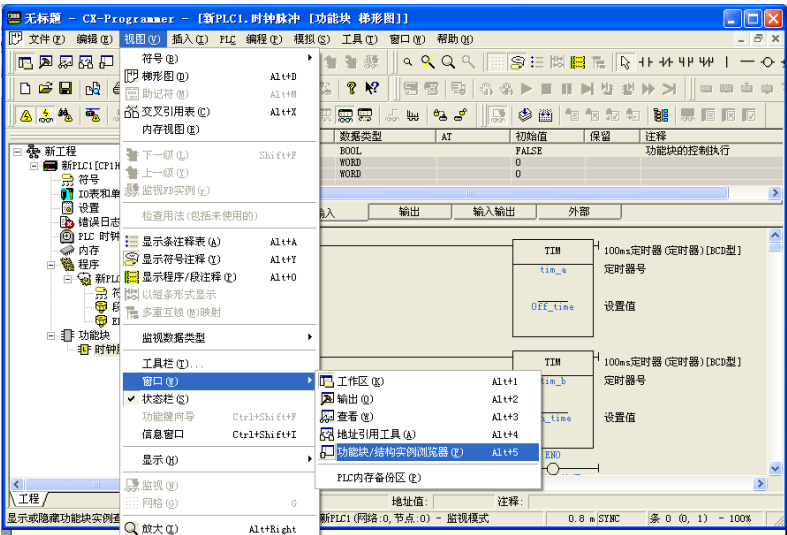


图 4.88 查看功能块实例的操作



图 4.89 功能块实例浏览器窗口

⑤ 5 s 后，定时器 `tim_a` 导通，则输出线圈 `ENO` 导通。

输出线圈 `ENO` 导通 2 s 后，定时器 `tim_b` 到时，切断 `tim_b` 常闭触点。定时器 `tim_a` 被重新复位，`tim_a` 触点断开，定时器 `tim_b` 被重新复位，输出线圈 `ENO` 断开。此时常闭触点 `tim_b` 导通，定时器 `tim_a` 又开始重新计时。这样，实现了输出线圈灭 2 s，亮 5 s 的操作。

如果需要输出为灭 3 s、亮 3 s，占空比为 11 的脉冲，则只须断开常开触点 0.00，接通 0.01。可见，如果不使用功能块方式编程则需要编写两套结构相同的脉冲产生程序（仅定时器的设定值不同），结构复杂，工作量大。因此功能块的使用简化了编程的步骤，程序结构更加清晰、简练，功能块适合于开发复杂的控制算法。

习 题

- 1. 什么系列的 PLC 没有手持编程器，只能通过 CX-P 软件进行编程？
- 2. 为什么 OMRON 公司开发了 CX-One 软件？CX-One 与 CX-P 有什么关系？
- 3. 在 CX-P 软件中，对 PLC 进行初始设置包括哪些内容？
- 4. 在 CX-P 软件中，输出窗口、查看窗口、地址引用工具和交叉引用表窗口各有什么用途？
- 5. 在 CX-P 软件中，全局符号与本地符号的区别是什么？如何设置？
- 6. 在工程工作区中，如何实现“PLC”、“PLC 程序”和“程序段”的添加？它们各有何用途？
- 7. 在对程序进行编译时，“错误”和“警告”信息有何区别？
- 8. “在线强制”功能在实际应用中有什么好处？
- 9. 在线修改程序的步骤是什么？
- 10. 简述创建功能块的步骤。
- 11. 如何实现在线监视功能块的功能？
- 12. 编写一功能块程序，实现对任意十进制数转换为十六进制数的功能。



# 第 5 章 任务编程方法

## 5.1 任务程序概述

以前的 OMRON PLC 程序是一条一条进行扫描执行的，在每一个扫描周期里，PLC 都要进行 I/O 刷新。因此，PLC 占用的扫描时间比较长。目前 OMRON 生产的主流 PLC：CP1 系列、CJ1 系列、CS1 系列 PLC，在程序上采用单元化结构，可以将程序按功能、控制对象、工序或者开发者等条件进行划分，分割成“任务”的执行单位。所谓任务就是规定使各个程序按照何种顺序或中断条件进行划分的功能。在任务中所分配的各程序是分别独立的，每个任务程序后需要有各自的 END 指令，如图 5.1 所示。

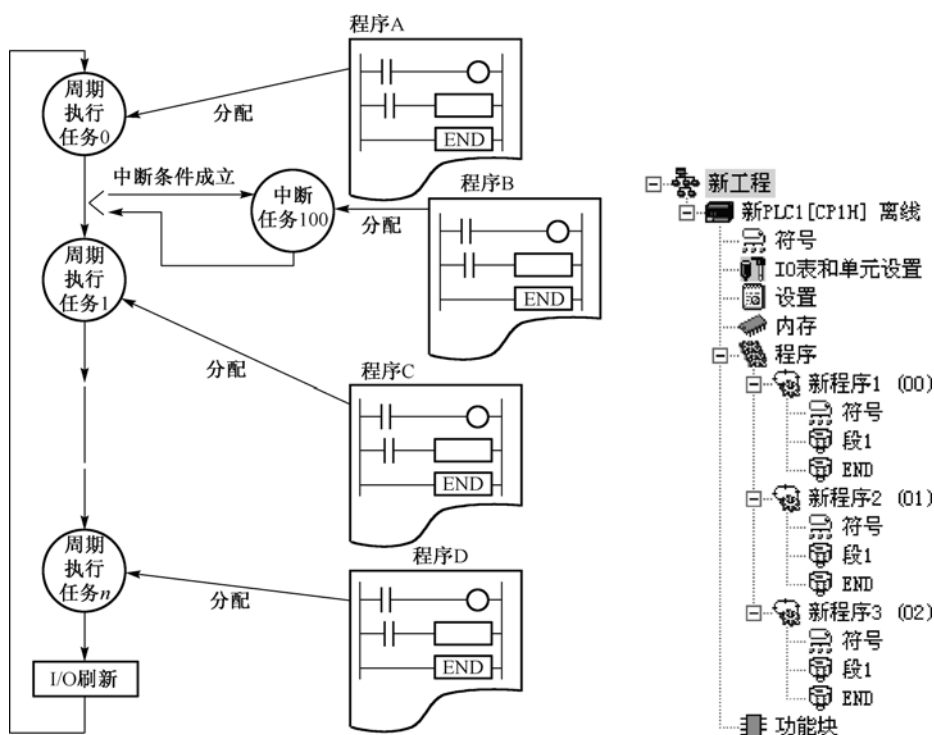


图 5.1 任务编程示意图

### 1. 任务编程的特点

由于任务编程是按功能、控制对象、工序或者开发者等条件进行划分，因此具有以下特点：

- ① 可将复杂程序分割后由多人共同开发，最后再将分割设计的各个程序段合并在一个用户程序中。
- ② 可将程序“定制”为标准化模块，特别是通过与外围工具的功能组合，程序可以不依赖于特定的系统，成为独立性很高的标准程序模块。其优势是能很方便地移植于其他系统，而且由多人开发的程序也很容易融合。

③ 提高总体的响应性能。将系统分为总体管理程序与独立控制程序，根据需要可以仅使某个控制程序进行动作，因此提高了系统总体的响应性能。

④ 修改与调试更加简便。由于可以由多人分工按任务进行编程、调试，提高了调试效率。即使 PLC 的规格改变时，只需对变更部分的任务进行修正即可，维护简便。通过 CX-Programmer (简称 CX-P) 软件中名称的全局/局部设定，以及“局部变量”地址的自动分配功能，在调试时能够很容易地识别某个地址是程序固有的还是与其他程序共享的。另外由于不需要进行程序间地址的重复检验，提高了调试效率。

⑤ 程序的内部处理更加简捷。在按用户机型对程序进行更改时，根据程序上的任务控制指令，能够按用户机型切换到所需执行的任务。

⑥ 增加用户程序的可读性。通过对用户程序进行结构化编写，对传统上采用转移指令的程序进行模块化处理，使用户程序更加容易理解。

## 2. 任务的种类

CP1H/CP1L 最多能够管理 288 个任务：32 个周期性任务+256 个中断任务 (CP1E 为 17 个任务：1 个周期任务+16 个中断任务)，每个任务由一段程序组成。任务大致可以分为两种：周期性执行任务和中断任务，其中中断任务可以作为追加任务来使用。

### (1) 周期性执行任务

周期性执行任务是指一个扫描周期内执行一次，即从第一逻辑行开始执行到 END 指令结束。最多能使用 32 个任务，按任务的顺序号 (No.0~31) 由小到大顺序执行。可以利用 CX-P 将程序的属性设定为“循环任务”或由 TKON 指令来调用。“循环任务”就是周期执行任务，“循环任务”的设定如图 5.2 所示。

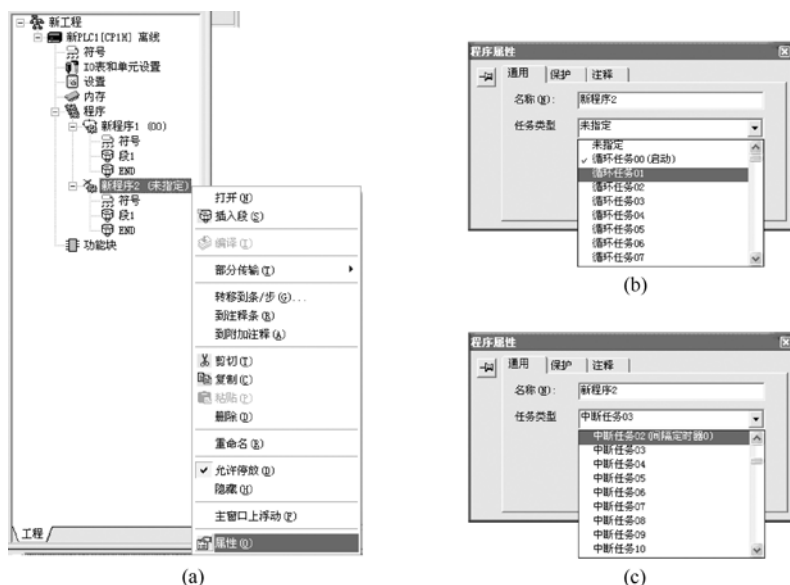


图 5.2 “循环任务”的设定

在 CX-P 编程软件的工程工作区，在“新 PLC[CP1H]”上单击右键，在快捷菜单中选择“插入程序”命令，在工程窗口得到添加的“新程序 2(未指定)”，右击“新程序 2(未指定)”，在快捷菜单中选择“属性”命令，如图 5.2(a) 所示，弹出“程序属性”对话框。在任务类型中选择“循环任务 01”，即完成将任务设定为“周期性执行任务”的过程。若在任务类型中选择“中断任务 00”~“中断任务 255”，如图 5.2(c) 所示，即完成将任务设定为“中断任务”的过程。

(2) 中断任务

中断任务是指当中断发生时，停止周期性执行任务/追加任务的执行，进行强制性中断，转而执行中断任务，执行完中断任务再返回中断前的断点继续执行中断前的任务。CPIH 的中断任务可分为 4 种：输入中断(直接模式、计数模式)、高速计数器中断、定时器中断和外部中断。最多有 256 个中断任务，编号为 No.0~255。“中断任务”的设置如图 5.2(a)、图 5.2(c) 所示。

(3) 追加任务

追加任务是设置了可执行任务状态的中断任务。追加任务能够和周期性执行任务一样周期性地运行中断任务。在运行完周期执行任务(周期执行任务 No.0~31)后，对追加任务(设置了可执行任务状态的中断任务)，按中断任务由小到大的顺序执行。最多有 256 个追加任务，编号为 No.0~255。但是，与周期执行任务不同的是，追加任务不具有“循环任务”的属性，不能将追加任务设置为进入运行模式直接执行，它只能由 TKON 指令来启动。

3. 任务的执行条件及其相关设定

任务的执行条件和相关设定见表 5.1。

表 5.1 任务的执行条件和相关设定

任 务 种 类		任 务 号	执 行 条 件	相 关 设 定
周期执行任务		0~31	在可执行状态下，取得执行权时，每个周期执行	无(总是有效)
中 断 任 务	输入中断 0~7	中断任务 140~147	CPU 单元内置输入点接通或接通达到一定次数时	由 MSKS(中断控制指令)设定中断的类型和中断号
	高速计数器中断	中断任务 0~255	CPU 单元内置高速计数器的目标值一致或区域比较的条件一致时执行	由 CTBL(比较表登录指令)设定比较类型和中断任务号
	定时器中断 0	中断任务 02	根据 CPU 单元内部定时器，每隔一定时间中断一次	由 MSKS(中断控制指令)设定中断类型和间隔时间，间隔时间单位由 CX-P 设定 10 ms/1.0 ms/0.1 ms
	外部中断	中断任务 0~255	来自扩展的 CJ 系列的高功能 I/O 单元或 CPU 高功能单元要求时	无
追加任务 0~255		中断任务 0~255	在可执行状态下，取得执行权时，每个周期执行	无

4. 周期性执行任务/追加任务的状态及转换

周期性执行任务/追加任务具有以下 4 个状态，根据条件对这 4 个状态进行转换。

(1) 不执行状态(INI)

不执行状态是指一次都未被执行的状态。在编程模式时所有的周期执行任务都为不执行状态。凡转化为其他状态的周期执行任务只要未切换为编程模式，就不能返回到该状态。

(2) 可执行状态(READY)

① 按照指令执行启动的任务，通过运行任务启动指令 TKON 将未执行状态或待机状态转换为可执行状态。

② 在运行开始时启动的任务(仅限周期执行任务)。从“程序”模式切换为“运行”模式或“监视”模式时，由不执行状态转化为可执行状态。选工程工作区的项目“新程序(0~31)”，单击快捷菜单的“属性”命令，可以将周期执行任务 No.0~31 中任意一个设置为“操作开始”，表示从运行开始时进入可执行状态。

(3) 执行状态(RUN)

当周期执行任务处于执行状态时获得执行权，处于实际执行的状态，即传统的程序执行状态。执行权按照该扫描周期内执行状态的任务由小到大的顺序依次传递。

(4) 待机状态 (WAIT)

根据任务执行待机 (TKOF) 指令，从执行状态切换为待机状态。在此状态下，指令不执行，因此不会增加指令的执行时间。

各任务状态的转换关系如图5.3所示。通过状态之间的转换，可以对不执行的程序区域进行任务分割，使之处于适当的待机状态，以缩短扫描周期。



图 5.3 4 种任务转换关系示意图

注：只能在周期执行任务中，将程序的属性设定为“操作开始”时启动。

5.2 任务的使用方法

1. 任务启动 TKON/任务待机 TKOF 指令

从程序上启动周期执行任务时，可使用任务启动 (TKON) 指令实现；而待机 (TKOF) 指令则将任务置于待机状态。

表 5.2 是任务控制类指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。

表 5.2 任务指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位的影响
任务启动指令		N: 任务 No. 周期执行任务 00~31 追加任务 8000~8255 (对应中断任务 000~255)	输入条件为 ON 时，将 N 所指定的周期执行任务置为可执行状态或将中断任务变为追加任务来执行 对标志位的影响：启动周期执行任务 00~31 对应的任务标志 (TK00~31) 置为 1
任务待机指令		N: 任务 No. 周期执行任务 00~31 追加任务 8000~8255 (对应中断任务 000~255)	输入条件为 ON 时，将由 N 所指定的周期执行任务或追加任务置为待机状态 对标志位的影响：对应的任务标志 (TK00~31) 置为 0

(1) 任务启动指令 TKON

任务启动指令 TKON 可使周期执行任务置为可执行状态或将中断任务变为追加任务。TKON N 的功能是由 N 所指定的周期执行任务或追加任务置为可执行状态。当 N=00~31 时，执行 TKON 指令时，对应的任务标志 (TK00~31) 置为 1。使用 TKON 指令置为可执行状态的周期执行任务或追加任务，只要 TKOF 指令不使之置为待机状态，在下一个周期仍保持为可执行状态，而且 TKON 指令可以在任何任务中设定其他任务。

注意：

- ① TKON 可以在周期执行任务或追加任务中执行，而不能在中断任务中执行。
- ② 当 TKON 指令将比自身任务号小的任务置为可执行状态时，该任务在本周期内不能执行，要到下一个周期方可执行；当 TKON 指令将比自身任务号大的任务置为可执行状态时，该任务在本周期内即被执行。

③ 对于任务标志已经置为 1 的任务而言, 执行 TKON 指令时无效, 视做空操作指令。而将自身任务号置为可执行状态时, 执行 TKON 指令也无效。

④ 在一个扫描周期中必须具有一个或一个以上的置为可执行状态的周期执行任务或追加任务。否则, 任务出错标志 A295.12 将置位, CPU 停止运行。

## (2) 任务待机指令 TKOF

TKOF N 指令的功能是将由 N 所指定的周期执行任务或追加任务置为待机状态。当 N=00~31, 执行 TKOF 指令时, 对应的任务标志(TK00~31)置为 0。使用 TKOF 指令置为待机状态的周期执行任务或追加任务, 只要 TKON 指令不使之置为可执行状态, 在下一个周期仍保持为待机状态。

注意:

① TKOF 可以在周期执行任务或追加任务中执行, 而不能在中断任务中执行。

② 当 TKOF 指令将比自身任务号小的任务置为待机状态时, 该任务将在下一个周期置为待机状态; 当 TKOF 指令将比自身任务号大的任务置为待机状态时, 该任务在本周期内即处于待机状态。

③ 若将自身任务置于待机状态, 在执行 TKOF 指令的同时, 本任务置为待机状态, 即在 TKOF 指令之后的指令不被执行。

## 2. 任务和 I/O 内存的关系

除了变址寄存器(IR)与数据寄存器(DR)外, 各任务将共享其他数据区域, 例如, 对于在周期执行任务 No.1 中使用的接点 10.00 和周期执行任务 No.2 中使用的接点 10.00 是指同一个点。因此对于 IR 和 DR 以外的内存区域, 由于是各任务共同使用导致了在某个任务中变更的值必然影响到其他任务, 所以在编程时需要特别注意。

对变址寄存器(IR)与数据寄存器(DR)而言, 在任务中有两种使用方法: 在各个任务分别(单独)使用的方法和各任务共同使用的方法。

① 在各个任务分别(单独)使用的方法: 在周期执行任务 1 中使用的 IR0 和在周期执行任务 2 中使用的 IR0 不相同。

② 各任务共同使用的方法: 在周期执行任务 1 中使用的 IR0 和周期执行任务 2 中使用的 IR0 为相同。对于变址寄存器(IR)及数据寄存器(DR)采用方法①还是采用方法②由 CX-P 进行设定。

如图 5.4 所示, 在 CX-P 编程软件的工程目录, 右键单击“新 PLC[CP1H]”, 在快捷菜单中选择“属性”命令, 在弹出的“PLC 属性”对话框的“通用”选项卡中, 选中“每个任务独立使用 IR/DRs”复选框, 如图 5.4(b)所示, 这样就采用了单独使用方法。

## 3. 任务对指令的使用限制

### (1) 不能跨任务使用的指令

以下配对使用的指令必须配置在同一任务之内, 否则错误标志 ER 置位(为 ON), 不能执行指令。

① JMP/JME 跳转/跳转结束

② CJP/JME 条件跳转/跳转结束

③ CJPN/JME 条件非跳转/跳转结束

④ JMP0/JME0 多重跳转/多重跳转结束

⑤ FOR/NEXT 重复开始/重复结束

⑥ IL/ILC 互锁/互锁清除

⑦ SBS/SBN/RET 子程序调用/子程序开始/子程序返回

⑧ MCRO/SBN/RET 宏/子程序开始/子程序返回

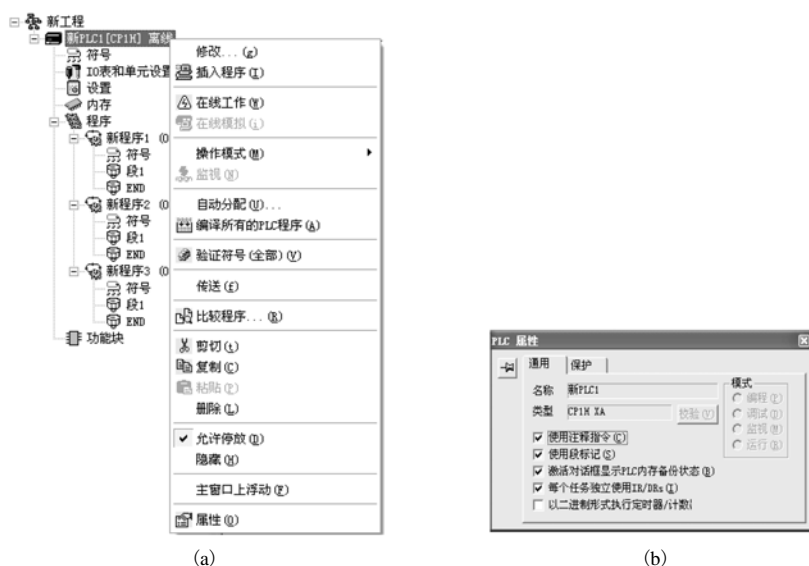


图 5.4 变址寄存器与数据寄存器的使用方法设置

⑨ BPRG/BEND 块程序/块程序结束

⑩ SNXT/STEP 步启动/步定义

注：本书没有提到的指令，详见 CP1H CPU 单元编程手册。

(2) 在中断任务内不能使用的指令

以下的指令在中断任务内不能执行。在中断任务内执行时 ER 标志为 ON，不能执行指令。但是把中断任务作为追加任务来使用时，能够使用以下的指令。

① TKON 任务启动

② TKOF 任务待机

③ SNXT 步启动

④ STEP 步定义

⑤ STUP 串行端口通信设定变更

⑥ DI 中断任务禁止

⑦ MSKR 中断读取

注：本书没有提到的指令，详见 CP1H CPU 单元编程手册。

(3) 在中断任务内不能保证动作的指令

以下所列指令在中断任务内执行时结果将不准确：

① TIM/TIMX 定时器

② TIMH/TIMHX 高速定时器

③ TMHH/TMHHX 超高速定时器

④ TTIM/TTIMX 累计定时器

⑤ MTIM/MTIMX 多输出定时器

⑥ TIML/TIMLX 长时间定时器

⑦ TIMW/TIMWX 块程序的定时器等待

⑧ TMHW/TMHWX 高速定时器等待

⑨ PID 控制

- ⑩ FPD 故障点检测
- ⑪ STUP 串行通信

注：本书没有提到的指令，详见 CP1H CPU 单元编程手册。

4. 任务标志

(1) 周期执行任务相关的标志

以下是与周期执行任务相关的标志，不适应追加任务。

① 任务标志 (TK00~TK31)：在确认当前任务是否被执行时使用。周期执行任务为可执行状态时，标志 TK00~TK31 为 1 (ON)，在不执行状态或待机状态时为 0。任务 No.00~31 对应于任务标志 TK00~TK31。

② 任务初次启动标志 (A200.15)：在运行中进行一次初始处理时使用。周期执行任务从不执行状态 (INI) 转换为可执行状态 (READY) 并得到执行权，处于初次执行状态时，A200.15 为 ON，初次执行结束时为 OFF，如图 5.5 所示。

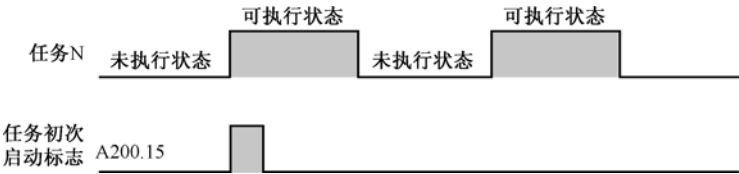


图 5.5 任务初次启动标志示例

③ 任务上升沿标志 (A200.14)：每次任务启动进行初始处理时使用。在周期执行任务从待机状态 (WAIT) 或未执行状态 (INI) 变成可执行状态 (READY) 时，A200.14 为 ON，与任务初次启动标志 (A200.15) 不同，如图 5.6 所示。

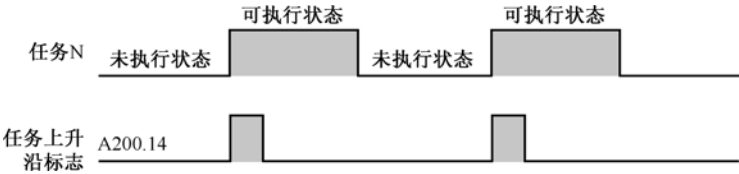


图 5.6 任务上升沿标志示例

(2) 任务的共享标志

① 任务出错标志 (A295.12)。

发生任务出错时，A295.12 为 1 (ON)。任务出错指下列情况之一：

- 可执行状态 (READY) 的周期执行任务/追加任务在该周期内一个也不存在。
- 不存在分配给周期执行任务的程序 (在使用外围工具 CX-P 时不发生这个状态)。
- 不存在分配给启动的中断任务 (包括追加任务) 的程序。

② 出错任务号通道 (A294 CH)。

由于程序出错而停止运行时，与出错任务种类和出错任务号有关的内容将保存在 A294 CH 通道中，见表 5.3。由此可以判断出错任务种类和发生异常的任务号。异常解除时 A294 CH 的值被清除。同时将该程序停止时所处的程序地址保存到 A298 CH (程序地址低字)、A299CH (程序地址高字) 中。

表 5.3 程序出错 A294 CH 内容的意义

任 务 种 类	A294 CH
周期执行任务	0000~001F Hex (对应任务 No.0~31)
中断任务 (包括追加任务)	8000~80FF Hex (对应中断任务 No.0~255)

5. 任务设计原则

任务设计需遵守以下原则。

① 按照基础条件分割任务，如图5.7所示。

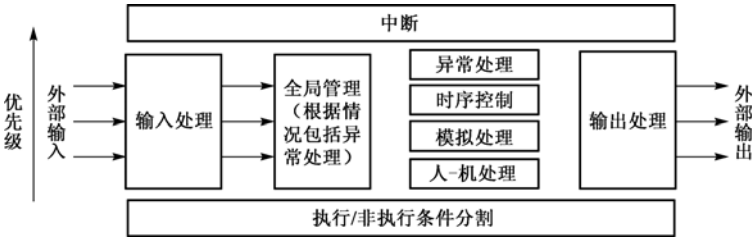


图 5.7 分割任务示意图

- 明确执行、非执行条件并汇总。
  - 按有无外部输入/输出汇总。
  - 按功能汇总。对于时序控制、模拟处理、人—机处理、异常处理等要尽可能地减少任务间的数据交换，从而提高独立性。
  - 按执行优先级汇总，分为通常任务和中断任务。
- ② 为提高任务的独立性，减少任务(程序)间的数据交换，要尽可能地进行分割设计。
- ③ 采用全体管理(任务控制)，用任务来控制各任务的执行或待机。
- ④ 为周期执行任务或追加任务中优先级高的任务分配小任务号。
- ⑤ 为优先级高的中断任务分配小中断任务号。

⑥ 任务一旦被启动，只要没有被自身或其他任务设定为待机状态，则在下一个周期之后成为可执行状态。执行按条件的任务分支处理时，不要忘记插入对于其他任务的 TKOF 指令。

⑦ 在进行任务执行时的初始化处理时，请使用以下标志：

- 运行中只进行一次初始处理时，使用任务初次启动标志(A200.15)。
- 只要任务启动就进行初始处理时，使用任务上升沿标志(A200.14)。

⑧ 划分清楚在各任务(程序)中共同使用的内存区域和只在各任务(程序)内所使用的内存区域，把只在各任务(程序)内使用的内存范围按各任务(程序)进行汇总。

5.3 中断任务

5.3.1 CP1 系列 PLC 的中断功能

所谓中断，是指在外部的或内部触发信号的作用下，中断周期执行程序(循环程序)的执行而转去执行一段预先编好的程序段，也称中断任务(或中断服务程序)，中断任务执行完毕再返回断点处继续执行周期执行程序的现象。

1. 中断任务的类型

CP1 系列 PLC 的中断功能比较完备，有 4 大类中断功能，分别是：输入中断(直接模式/计数器模式)、高速计数器中断、间隔定时器中断和外部中断。

(1) 输入中断(直接模式/计数器模式)

外部发生的事件所产生的信号通过中断输入点(0.00~0.03、1.00~1.03)送入 CP1H X/XA 型 PLC，当某个中断输入点为 ON(输入中断直接模式)或 ON 一定次数时(输入中断计数器模式)，产生中断请求信号。所对应的中断任务 140~147。若不使用中断功能时，这些点可以作为普通输入点使用。



(2) 高速计数器中断

CP1H CPU 单元内置的高速计数器对输入脉冲进行计数，计数器当前值与预先登录的目标值比较表或区域比较表条件满足时，可使指定的中断任务 0~255 (每个任务是由一段程序组成) 启动。

(3) 间隔定时器中断

间隔定时器中断 (简称定时器中断) 通过 CPU 单元的内置定时器，按照一定的时间间隔执行中断任务的处理。中断任务 2 被固定分配为间隔定时器中断。间隔定时器中断不占用 CP1H 的输入点。

(4) 外部中断

外部中断来自扩展的 CJ 系列的高功能 I/O 单元或 CPU 高功能单元。

CP1H X/XA 型 PLC 中断任务见表 5.4。

表 5.4 CP1H X/XA 型 PLC 中断任务一览表

中 断 类 型	中断任务号	中 断 条 件	设 定 方 法	最 大 点 数
输入中断 0~7	140~147	直接模式：中断输入点接通 计数器模式：中断输入点的接通达到一定次数	由 MSKS (中断控制指令) 设定中断的类型和中断号	8 点
高速计数器中断	0~255	内置高速计数器的计数当前值与目标值比较表或区域比较表的比较条件满足时	由 CTBL (比较表登录指令) 设定比较类型和中断任务号	256 点
间隔定时器中断	2	根据 CPU 单元内部定时器，每隔一定的时间中断一次	由 MSKS (中断控制指令) 设定中断类型和间隔时间 间隔时间单位由 CX-P 设定	1 点
外部中断	0~255	来自扩展 CJ 系列的高功能 I/O 单元或 CJ 系列 CPU 高功能单元要求时	无 (总是有效)	256 点

CP1 系列 PLC 的任务功能列于表 5.5，以便读者了解。

表 5.5 CP1 系列 PLC 的任务功能

任务种类 \ 机型		CP1H		CP1L			CP1E		
		X/XA	Y	20~60 点	14 点	10 点	20~60	14 点	10 点
周期执行任务		0~31 (32 个)		0~31 (32 个)			1 个	1 个	1 个
输入中断输入点		0.00~0.03, 1.00~1.03		0.04~0.09			0.02~0.07		
中 断 任 务	输入中断号/ 中断任务号	0~7/ 140~147	0~5/ 140~145	0~5/ 140~145	0~3/ 140~143	0~1/ 140~141	2~7/ 2~7	2~7/ 2~7	2~5/ 2~5
	高速计数器中断	中断任务 0~255		中断任务 0~255			中断任务 0~15		
	定时器中断 0	中断任务 02		中断任务 02			中断任务 01		
	外部中断	0~255	0~255	无	无	无	无	无	无

2. 中断的优先级

CP1H 在执行某中断任务 A 时，其他中断 B 发生，则要等 A 的处理结束后 B 的中断处理才能开始。CP1H 在同时发生多个中断任务时，中断的优先级如图 5.8 所示。

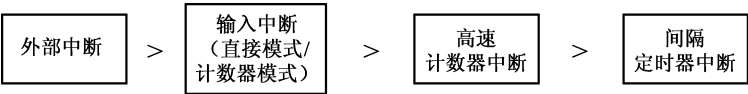


图 5.8 CP1H 的中断优先级

在同一类中断中，若几个中断同时发生请求，中断任务号小的优先级别高。例如，输入中断中几个中断输入点同时为 ON 时，则执行中断的优先顺序为：中断输入 0→中断输入 1→中断输入 2→中断输入 3→……。

3. 中断任务的设定

在 CX-P 编程软件的工程目录中，右键单击“新 PLC[CP1H]”，在快捷菜单中选择“插入程序”命令，在工程窗口得到添加的“新程序 2(未指定)”，右击“新程序 2(未指定)”，在快捷菜单中选择“属性”命令，如图 5.9(a)所示，弹出“程序属性”对话框。在任务类型中选择“中断任务 00”～“中断任务 255”，即完成将任务设定为“中断任务”的过程。若在任务类型中选择“循环任务 00”～“循环任务 31”，即完成将任务设定为“循环任务”的过程。



图 5.9 中断任务的设定

4. 中断控制类指令

下面以 CP1H X/XA 型 PLC 为例，介绍中断控制类指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响，见表 5.6。

表 5.6 中断控制类指令

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位影响
中断控制指令 MSKS/@MSKS	<div><div>MSKS(690) N S</div><div>@MSKS(690) N S</div></div>	<p>N: 中断类型定义，如下所示 S: 控制数据，范围为 CIO、W、H、T/C、A、D、@/*D、DR 等。依 N，S 意义如下：</p> <div><div>@MSKS(690) 110~117--</div><div>输入中断0~7 S----- 0000: 上升沿执行中断           0001: 下降沿执行中断 110~113对应可写为10~13</div><div><div>@MSKS(690) 100~107--</div><div>输入中断0~7 S----- 0000: 中断允许           0001: 中断禁止           0002: 中断允许，减计数器           0003: 中断允许，加计数器 100~103对应可写为6~9</div><div><div>@MSKS(690) N----- S</div><div>14: 定时器中断复位定时 4: 定时器中断不复位定时 由CLi设定初次中断时间 ----- 设定时间(1~9999),           设定时间为0时禁止中断</div></div></div></div>	<p>执行条件为 ON 时，将按设置执行相应的输入中断或定时中断 对标志位的影响：N 或 S 的数据不在指定范围，ER 为 ON</p>

(续表)

名    称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位影响
中断读取指令 MSKR/@MSKR	<div><div>MSKR(692)</div><div>N</div><div>D</div></div> <div><div>@MSKR(692)</div><div>N</div><div>D</div></div>	N: 中断类型定义 D: 输出通道 范围: CIO、W、H、T/C、A448~959、D、@/*D、DR 等 其他同上 4: 读取设定值 14: 读取当前值	执行条件为 ON 时, 读取通过 MSKS 指令指定的中断控制的设置, 存在 D 中。两种情况: ① 输入中断时, 读取 MSKS 指令的第 2 个操作数 S, 存在 D 中 ② 定时器中断 N=4 时, 以十六进制形式输出定时器的设定值, 存在 D 中; N=14 时以十六进制形式输出定时器的当前值, 存在 D 中 对标志位的影响: N 的数据不在指定范围, ER 为 ON
中断清除指令 CLI/@CLI	<div><div>CLI(690)</div><div>N</div><div>S</div></div> <div><div>@CLI(690)</div><div>N</div><div>S</div></div>	N: 中断类型定义。S 意义如下: ① N=100~107 时, 为输入中断 0~7, 对应中断任务 No.140~147 S=0: 记忆保持; S=1: 记忆清除 ② N=10~13 时, 为高速计数器中断 0~3 S=0: 记忆保持; S=1: 记忆清除 ③ N=4 时, 为定时器中断初次中断开始时间的设定, S 为设定时间的保存通道 S: 设定时间通道, 范围为 CIO、W、H、T/C、A、D、@/*D、DR 等	执行条件为 ON 时, 对输入中断、高速计数器中断所设置信息进行清除或保持; 或进行定时器中断初次中断开始时间的设定(对应 MSKS 指令, N=4 时的使用), S 为设定时间的保存通道。因此, 要使输入中断任务执行过程中记忆的中断无效时, 可通过 CLI 指令, 清除被记忆到内部的中断 对标志位的影响: N 的数据不在指定范围, ER 为 ON
中断任务禁止指令 Di/@Di	<div><div>Di(693)</div></div> <div><div>@Di(693)</div></div>	无	执行条件为 ON 时, 禁止执行所有中断任务 对标志位的影响: 中断任务内执行时, ER 为 ON
解除中断任务禁止指令 Ei/@Ei	<div><div>Ei(694)</div></div> <div><div>@Ei(694)</div></div>	无	执行条件为 ON 时, 解除通过 Di 指令设定的所有中断任务执行的禁止 对标志位的影响: 同上

5. 指令的解释

(1) MSKS 指令

MSKS 指令的操作数有两套意义, 有两组功能。

① 输入中断。

N=110~117 (110~113 可对应写为 10~13) 时, 为输入中断 0~7 的直接模式, 对应中断任务 No.140~147。

S: 0000 指定上升沿中断, 0001 指定输出的下降沿中断。

N=100~107 (100~103 可对应写为 6~9) 时, 也为输入中断 0~7, 对应中断任务 No.140~147。但这时 S 的含义不同。

S: 0000 允许中断, 0001 禁止中断。

0002 允许中断, 计数器减法计数一定值, 则开始中断, 为减计数器模式。

0003 允许中断, 计数器加法计数一定值, 则开始中断, 为加计数器模式。

② 间隔定时器中断(中断任务 2)。

N=14 时, 为定时器中断(中断任务 2), 表示复位后开始定时, 每一定时间中断 1 次。

N=4 时, 为定时器中断(中断任务 2), 表示非复位定时, 每一定时间中断 1 次。要另行通过 CLI 指令指定初次中断开始时间, 内部定时器将从当前值定时, 第 1 次中断的时间由 CLI 指令指定, 如果无 CLI 指令指定, 定时开始到初次中断任务之间的时间无法确定。2 次中断后, 将按定时器设定的时间间隔定时启动中断任务。

(2) 中断清除 CLI 指令

CLI 指令用于进行输入中断、高速计数中断、记忆的清除/保持, 以及定时中断的初次中断开始时间的

设定。在执行输入中断任务过程中，若接收了其他编号的中断输入，则该编号中断被记忆到内部(只要不执行中断清除 CLI 指令)，此后被记忆的中断任务将按编号从小到大的顺序执行。因此，要使输入中断任务执行过程中记忆的中断任务无效时，可通过 CLI(中断清除)指令，清除被记忆到内部的中断。

5.3.2 输入中断

1. 输入中断的输入点

CP1H 系列 X/XA 型的 PLC 可将 0.00~0.03(输入中断 0~输入中断 3)、1.00~1.03(输入中断 4~输入中断 7)共 8 个输入点作为输入中断使用。Y 型的 PLC 可将 0.00~0.01(输入中断 0~输入中断 1)、1.00~1.03(输入中断 4~输入中断 7)共 6 个输入点作为输入中断使用。CP1L 和 CP1E 系列 PLC 分别可将 0.04~0.09 和 0.02~0.07 共 8 个输入点作为输入中断使用，CP1L/CP1E 点数规格不同，中断输入个数不同，详细情况请查相关手册。

外部发生的事件所产生的信号通过中断输入点送入 PLC，当某个中断输入点为 ON(输入中断直接模式)、或 ON 一定次数时(输入中断计数器模式)，产生中断请求信号。若不使用中断功能时，这些点可以作为普通输入点使用。

2. 输入中断的模式

输入中断有直接模式和计数器模式两种模式。

(1) 直接模式

在非屏蔽情况下，只要中断输入点接通则产生中断响应，执行相应中断任务 140~147。

若在屏蔽情况下，即使中断输入点接通也不能产生中断响应，但该中断信号被记忆下来，待屏蔽解除后立即产生中断。若屏蔽解除后不希望响应所记忆的中断，可用指令清除该记忆。

(2) 计数器模式

这种模式的中断，是对中断输入点接通的次数进行计数，当达到设定的次数时启动相应的中断任务，计数器模式的输入中断与直接模式的输入中断输入端子相同。

① 计数方法可通过 MSKS 指令设定，可选择加法模式和减法模式。

② 通过输入中断(计数器模式)启动的中断任务 No.与输入中断(直接模式)相同，都是中断任务 140~147。

③ 输入中断(计数器模式)计数频率最高为 5 kHz 以下。

3. 输入中断(计数器模式)的计数器区域

对计数器模式的输入中断，CP1H 系列有存放计数器设定值和当前值的区域。中断输入点、中断任务 No.及与计数器区域的关系见表 5.7。

表 5.7 输入中断(计数器模式)的计数器区域

中断输入点		功 能		计 数 器	
X/XA 型	Y 型	输入中断编号	中断任务 No.	设定值通道	当前值通道
0.00	0.00	输入中断 0	140	A532	A536
0.01	0.01	输入中断 1	141	A533	A537
0.02	—	输入中断 2	142(Y 型不可用)	A534	A538
0.03	—	输入中断 3	143(Y 型不可用)	A535	A539
1.00	1.00	输入中断 4	144	A544	A548
1.01	1.01	输入中断 5	145	A545	A549
1.02	1.02	输入中断 6	146	A546	A550
1.03	1.03	输入中断 7	147	A547	A551

4. CP1H 输入中断的设定

CP1H 在使用输入中断之前，要通过 CX-P 进行设定，输入中断设定分为以下两步：

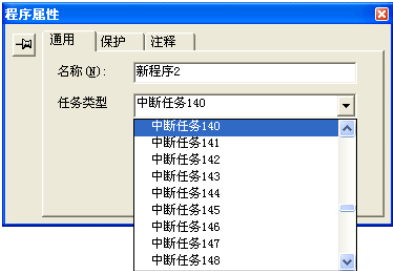


图 5.10 “程序属性”对话框

① 在 CX-P 编程软件的工程目录下，右击“新 PLC[CP1H]”，选择“插入程序”命令，则在工程窗口得到添加的“新程序 2(未指定)”，右击“新程序 2(未指定)”，选择“属性”命令，弹出“程序属性”对话框，如图 5.10 所示，在“通用”选项卡的任务类型中选择中断任务编号，例如选择“中断任务 140”，即选择输入点 0.00 对应的中断输入 0。

② 在 CX-P 编程软件的工程目录下，双击项目树的“设置”，弹出“PLC 设定”对话框，如图 5.11 所示。在“内置输入设置”选项卡中，最下方的选项为“中断输入”，将“IN0”设置为“中断”，以对应“中断任务 140”。



图 5.11 “PLC 设定”对话框

5. 应用示例

例 5-3-1 图 5.12 是输入中断(直接模式)的例子。将输入 0.00 置于 ON 时，希望执行中断任务 140。设定：

- ① 将输入设备连接到输入点 0.00。
- ② 在 CX-P 编程软件的工程目录下，双击项目树的“设置”，弹出“PLC 设定”对话框，在“内置输入设置”选项卡中，把最下方的“中断输入”选项的“IN0”选为“中断”。
- ③ 通过 CX-P 添加“中断任务 140”的新程序，编制中断处理用的程序。
- ④ 通过 CX-P 编制程序，如图 5.12(a) 所示。

图 5.12(a) 中，用了两条 MSKS 指令，完成了 MSKS 指令的两个功能。第 1 条 MSKS 指令的第 1 个操作数为 110，指的是输入点 0.00 作为中断输入点，对应中断任务 140；第 2 个操作数是 0000，表示中断是在输入点 0.00 的上升沿执行。第 2 条 MSKS 指令的第 1 个操作数为 100，指的是输入点 0.00 作为中断输入点，对应中断任务 140；第 2 个操作数是 0000，表示允许中断。

两条指令组合完成中断任务：执行条件的 W0.00 为 ON 时，通过 MSKS 指令的执行，可对 0.00 的上升沿进行输入中断动作。如输入 0.00 从 OFF 向 ON 变化(上升沿)，则将执行中的周期执行任务的

处理暂时中断，开始中断任务 140 的处理。如中断任务的处理结束，则再次开始已中断的梯形图程序的处理。动作过程如图 5.12 (b) 所示。

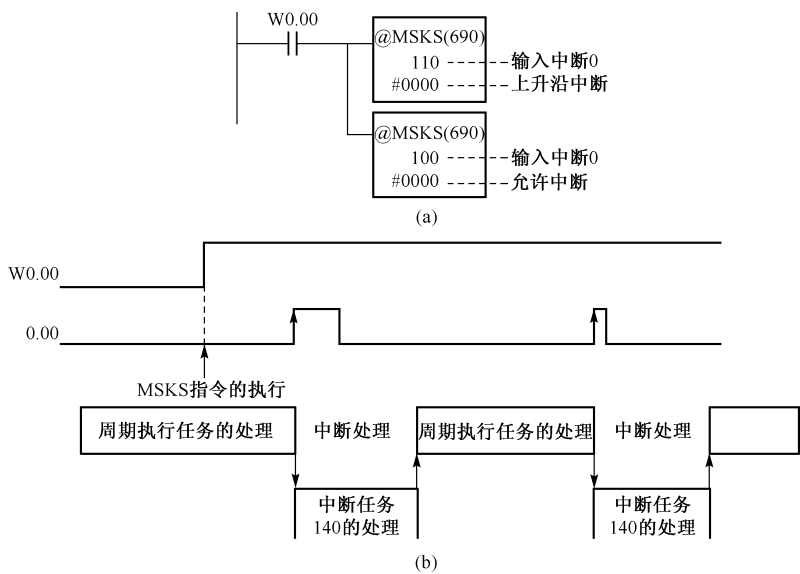


图 5.12 输入中断(直接模式)示例

**例 5-3-2** 图5.13是输入中断(计数器模式)的例子。对输入 0.01 的上升沿完成 200 次计数时，转去执行中断任务 141，要求计数方式设置为加法计数器方式。

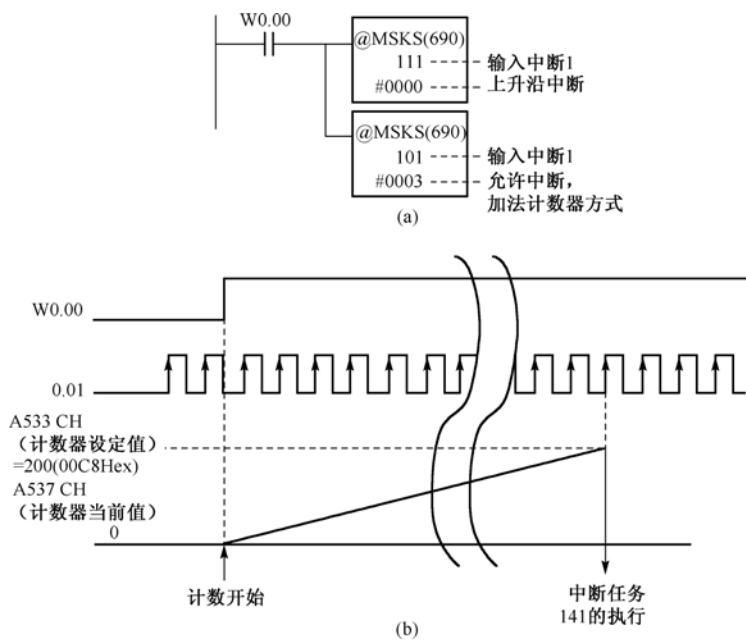


图 5.13 输入中断(计数器模式)示例

- 设定：
- ① 将输入设备连接到输入点 0.01。
  - ② 在 CX-P 编程软件的工程目录下，双击项目树的“设置”，弹出“PLC 设定”对话框，在“内置输入设置”选项卡中，“IN1”选为“中断”，即将输入点 0.01 设置为中断输入点。

- ③ 通过 CX-P 添加中断任务 141 的新程序，编制中断处理用的程序。
- ④ 通过 CX-P 将中断计数器的设定值 00C8Hex (200 次计数) 设定到 A533 中。
- ⑤ 通过 CX-P 编制程序，如图 5.13(a) 所示。

图 5.13(a) 中，用了两条 MSKS 指令，完成了 MSKS 指令的两个功能。第 1 条 MSKS 指令的第 1 个操作数为 111，指的是输入点 0.01 作为中断输入点，对应中断任务 141；第 2 个操作数是 0000，表示中断是在输入点 0.01 的上升沿执行。第 2 条 MSKS 指令的第 1 个操作数为 101，指的是输入点 0.01 作为中断输入点，对应中断任务 141；第 2 个操作数是 0000，表示允许中断，加法计数器方式。动作过程如图 5.13(b) 所示。

两条指令组合完成中断任务：执行条件 W0.00 变为 ON 时，可进行输入中断(计数器模式)的动作。如输入 0.01 进行 200 次 ON，则将执行中的周期执行任务的处理暂时停止，开始中断任务 141 的处理。如中断任务的处理结束，则再次开始已中断的梯形图程序的处理。

5.3.3 高速计数器中断

普通计数器 CNT 的计数脉冲频率受扫描周期及输入滤波器时间常数的限制，所以不能对高频脉冲信号进行计数。对高频脉冲信号的计数，大、中型 PLC 采用特殊功能单元来处理。对小型 PLC，如 CPH 系列 PLC 等，由于其设置了高频脉冲信号的输入点，配合相关的指令及必要的设定，也可以处理高频脉冲信号的计数问题。本节介绍 CPH 系列 PLC 的高速计数器及其功能。

1. 高速计数器的计数功能

(1) 高速计数器脉冲的输入端分配

CPH X/XA 系列 PLC 有 4 个高速计数器，可以分别对以下点输入的高速脉冲进行计数，即分别占用下述各点。

高速计数器 0，输入端 0.03、0.08、0.09(分别为 Z 相/复位、A 相/加法/计数输入、B 相/减法/方向输入)；高速计数器 1，输入端 0.02、0.06、0.07(分别为 Z 相/复位、A 相/加法/计数输入、B 相/减法/方向输入)；高速计数器 2，输入端 0.01、0.04、0.05(分别为 Z 相/复位、A 相/加法/计数输入、B 相/减法/方向输入)；高速计数器 3，输入端 1.00、0.10、0.11(分别为 Z 相/复位、A 相/加法/计数输入、B 相/减法/方向输入)。

当不使用高速计数器时，这些点可做普通输入点使用。

(2) 高速计数器的输入模式

PLC 在进行高速计数时，有时会用到旋转编码器。旋转编码器能输出脉冲信号，高速计数器配合旋转编码器使用，可以用于测量、处理转动或位移信号等。

不同型号的旋转编码器输出的脉冲也不相同，有的旋转编码器能产生单相脉冲信号，对应每个脉冲信号的前沿，高速计数器计数；有的旋转编码器能产生相位差为 90° 的两相脉冲信号，对应每个脉冲信号的前沿和后沿，高速计数器计数，至于 A 相和 B 相脉冲谁超前、谁滞后，这取决于旋转编码器的旋转方向。有的旋转编码器还能产生一个复位 Z 信号。

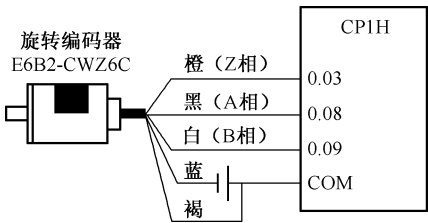


图 5.14 旋转编码器与 CPH 高速计数器 0 的连接

旋转编码器与 CPH 的高速计数器 0 连接如图 5.14 所示。  
CPH 高速计数器的输入模式有 4 种：递增模式、相位差模式、加/减模式和脉冲+方向模式。

① 递增模式。以高速计数器 0 为例，递增模式的脉冲信号由 0.08 输入。递增模式的最高计数频率是 100 kHz。在输入脉冲信号的上升沿，高速计数器的当前值加 1。

递增模式的脉冲输入信号如图5.15所示。

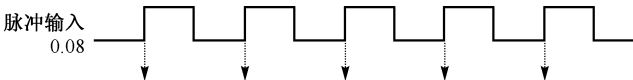


图 5.15 递增模式的脉冲输入信号

② 相位差模式。

以高速计数器 0 为例，相位差模式的 A 相脉冲由 0.08 输入，B 相脉冲由 0.09 输入，复位 Z 相信号由 0.03 输入。相位差模式计数的最高频率是 50 kHz。

相位差模式的输入信号如图5.16所示。

当 A 相脉冲超前 B 相脉冲 90°时，在 A、B 相脉冲的上升沿和下降沿处，计数器的当前值加 1。

当 B 相脉冲超前 A 相脉冲 90°时，在 A、B 相脉冲的上升沿和下降沿处，计数器的当前值减 1。

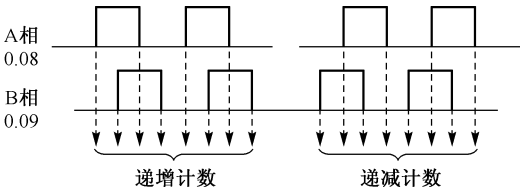


图 5.16 相位差模式的输入信号

③ 加/减模式。以高速计数器 0 为例，CW(顺时针方向)脉冲由 0.08 输入，CCW(逆时针方向)脉冲由 0.09 输入。加/减模式的最高计数频率是 100 kHz。

加/减模式的输入信号如图5.17所示。

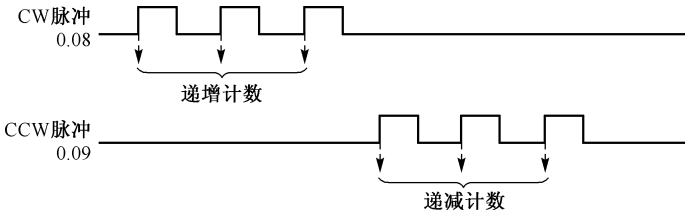


图 5.17 加/减模式的输入信号(CW：顺时针，CCW：逆时针)

使用加/减模式时，当 CW 信号出现时递增计数，当 CCW 信号出现时递减计数。在 CW 脉冲的上升沿，高速计数器的当前值加 1。在 CCW 脉冲的上升沿，高速计数器的当前值减 1。

④ 脉冲+方向模式。以高速计数器 0 为例，计数脉冲信号由 0.08 输入，方向信号由 0.09 输入。脉冲+方向模式的最高计数频率为 100 kHz。

脉冲+方向模式的输入信号如图5.18所示，当方向信号 OFF 时递减计数，当方向信号 ON 时递增计数。

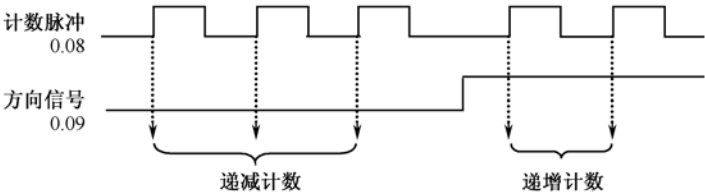


图 5.18 脉冲+方向模式的输入信号



### (3) 高速计数器计数模式

高速计数器计数模式有线形模式和循环模式两种模式。

① 线形模式。是对从下限值到上限值范围内的输入脉冲进行计数。若输入脉冲超过此上下限，则发生溢出/下溢，停止计数动作。

递增模式：数值范围为 00000000Hex~FFFFFFFFHex 或 0~4294967295(十进制)。

增/减模式：数值范围为 80000000Hex~7FFFFFFFHex 或 -2147483648~+2147483647(十进制)。

当高速计数器计数时，若从上限值开始进行递增计数就会发生上溢出，发生溢出时，其当前值为 0FFFFFFF；若从下限开始进行递减计数就会发生下溢出，发生溢出时，其当前值为 FFFFFFFF。发生溢出时计数器停止计数。重新复位高速计数器时，将清除溢出状态。

② 循环模式。在设定范围(0~最大值)内对输入脉冲进行循环计数。循环模式的最大值可以通过在 CX-P 的工程目录下，双击项目“设置”，弹出的“PLC 设定”对话框在“内置输入设置”选项卡中，选“循环模式”，可在 00000001~FFFFFFFFHex 的范围内任意设定。

### (4) 高速计数器的复位方式

高速计数器复位时，其当前值 PV = 0。CP1H 系列 PLC 的高速计数器本质上有两种常用的复位方式，可扩展为 4 种复位方式：(硬件复位)Z 相信号+软件复位、软件复位、(硬件复位)Z 相信号+软件复位(重启)、软件复位(重启)。

① Z 相和软件复位。当高速计数器的复位标志位 A531.00~A531.03(分别为高速计数器 0~3 的复位标志位)为 ON 且 Z 相信号也为 ON 时，高速计数器复位。

② 软件复位。当高速计数器的复位标志位 A531.00~A531.03(分别为高速计数器 0~3 的复位标志位)为 ON 时，高速计数器复位。

高速计数器复位时，其当前值 PV 为 0。

将高速计数器复位时，可通过 PLC 系统设定选择停止或重启比较动作。这样复位时，可进行将计数器从 0 状态开始再启动比较动作的应用。所以得到另外两种复位方式。

③ Z 相和软件复位(重启比较)。当高速计数器的复位标志位 A531.00~A531.03(分别为高速计数器 0~3 的复位标志位)为 ON 且 Z 相信号也为 ON 时，高速计数器复位。复位时，其当前值 PV 为 0。之后可进行将计数器从 0 状态开始再启动比较动作的应用。

④ 软件复位(重启比较)。当高速计数器的复位标志位 A531.00~A531.03(分别为高速计数器 0~3 的复位标志位)为 ON 时，高速计数器复位。复位时，其当前值 PV 为 0。之后可进行将计数器从 0 状态开始再启动比较动作的应用。

### (5) 高速计数器的当前值存储区(详见 OMRON CP1H CPU 单元操作手册)

对 CP1H 系列 PLC，4 个高速计数器的当前值放在 A 区的不同通道中。

高速计数器 0 的当前值存于 A271(存高 4 位)和 A270(存低 4 位)；

高速计数器 1 的当前值存于 A273(存高 4 位)和 A272(存低 4 位)；

高速计数器 2 的当前值存于 A317(存高 4 位)和 A316(存低 4 位)；

高速计数器 3 的当前值存于 A319(存高 4 位)和 A318(存低 4 位)。

使用计数器功能时，对应的通道已经被占用时，不能再作他用。

### (6) 高速计数器的设定

使用高速计数器时，部分内容要预先在 CX-P 编程软件上进行设置，否则高速计数器是不工作的。需要设置的内容见表 5.8。

表 5.8 PLC 系统设定的设定内容

项 目	设 定 内 容
高速计数器 0~3 的使用	使用
计数模式	线形模式
	循环模式
循环计数器最大值	0~FFFFFFF (Hex) (选择循环计数器时, 设定最大值)
复位方式	Z 相信号+软复位
	软复位
	Z 相信号+软复位 (重启比较)
	软复位 (重启比较)
输入模式	增量脉冲输入 (递增模式)
	相位差输入 (4 倍频)
	加/减输入
	脉冲+方向输入

在 PLC 的 CX-P 编程软件的工程窗口目录下, 双击“设置”, 弹出“PLC 设定”对话框, 如图 5.19 所示。在“内置输入设置”选项卡中, 有高速计数器 0~高速计数器 3 的选项: 使用高速计数器、计数模式、复位、输入设置。依次选择以确定使用的高速计数器、计数模式、复位方式及输入模式等。本例中, 选择使用高速计数器 0、增量脉冲输入模式(递增模式)、线形计数模式, 复位方式采用 Z 相信号+软复位。



图 5.19 “PLC 设定”对话框

2. 高速计数器的中断功能

配合一定的指令、进行必要的设定时, 高速计数器能实现中断功能。

CP1H CPU 单元内置的高速计数器当前值与预先写入的比较数据一致时, 可使指定的中断任务 0~255 (每个任务是由一段程序组成) 启动。

中断程序的结构和执行流程如图 5.20 所示。中断功能具有非常重要的意义, 因为在实际控制过程

中，控制系统中有些随时可能发生的情况需要 PLC 处理，具有中断功能的 PLC 可以不受扫描周期的影响，及时地把这种随机的信息输入到 PLC 中，从而提高了 PLC 对外部信息的响应速度。

高速计数器有两类中断方式，即目标值比较中断和区域比较中断，如图 5.21 所示。

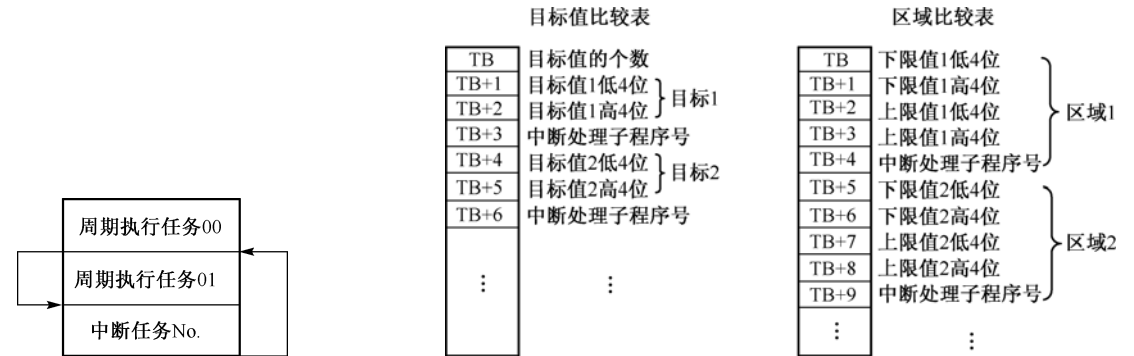


图 5.20 中断任务示意图

图 5.21 两种比较表的结构

(1) 目标值比较中断

在采取目标值比较中断时，要建立一个目标值比较表，如图 5.21 (a) 所示。目标值比较表占用一个区域的若干个通道，其中首通道存放目标值个数(BCD 数)。比较表中最多放 48 个目标值，每个目标值占两个通道(各存放目标值的低 4 位和高 4 位)，加上每个目标值对应一个中断任务号，存放 48 个子程序号需要 48 个通道，所以目标值比较表最多占用 145(48×3+1)个通道。目标值比较表中的数据可用编程软件预先写入。

目标值比较中断的执行过程是：在高速计数器计数过程中，若其当前值与比较表中某个目标值相同时，则停止执行主程序而转去执行与该目标值对应的中断任务。中断任务执行完毕，返回到断点处继续执行主程序。

(2) 区域比较中断

在采取区域比较中断时，要建立一个区域比较表，如图 5.21 (b) 所示。区域比较表分为 8 个区域，每个区域占 5 个通道，其中两个通道用来存放下限值的低 4 位和高 4 位、两个通道用来存放上限值的低 4 位和高 4 位，一个通道存放与该区域对应的中断任务号。8 个区域要占 40 个通道。当实际使用的比较区域不满 8 个时，要把其余区域存放上、下限值的通道都置为 0，将存放中断任务号的通道都置为 FFFF。区域比较表中的数据可用编程软件预先写入。

区域比较中断的执行过程是：在高速计数器计数过程中，若其当前值落在区域比较表中某个区域时，即下限值≤高速计数器 PV 值≤上限值，则停止执行周期执行程序而转去执行与该区域对应的中断任务。中断任务执行完毕，返回到断点处继续执行周期执行程序。

执行区域比较中断时，4 个高速计数器的比较结果存放在 A274、A275 和 A320、A321 中。

高速计数器 0 的区域比较结果存于 A274(区域 1 对应 A274.00～区域 8 对应 A274.07)；

高速计数器 1 的区域比较结果存于 A275(区域 1 对应 A275.00～区域 8 对应 A275.07)；

高速计数器 2 的区域比较结果存于 A320(区域 1 对应 A320.00～区域 8 对应 A320.07)；

高速计数器 3 的区域比较结果存于 A321(区域 1 对应 A321.00～区域 8 对应 A321.07)。

详细内容见欧姆龙 CP1H CPU 单元操作手册。

(3) 高速计数器的状态区

高速计数器的状态区是指高速计数器的各种信息存放的指定数据区，查看它们可以了解高速计数器的工作情况。高速计数器的状态区见表 5.9。

表 5.9 高速计数器的状态区

内 容		高速计数器 0	高速计数器 1	高速计数器 2	高速计数器 3
当前值保存区域	保存高位 4 位	A271 CH	A273 CH	A317 CH	A319 CH
	保存低位 4 位	A270 CH	A272 CH	A316 CH	A318 CH
区域比较一致标志	与比较条件 1 相符时为 ON	A274.00	A275.00	A320.00	A321.00
	与比较条件 2 相符时为 ON	A274.01	A275.01	A320.01	A321.01
	与比较条件 3 相符时为 ON	A274.02	A275.02	A320.02	A321.02
	与比较条件 4 相符时为 ON	A274.03	A275.03	A320.03	A321.03
	与比较条件 5 相符时为 ON	A274.04	A275.04	A320.04	A321.04
	与比较条件 6 相符时为 ON	A274.05	A275.05	A320.05	A321.05
	与比较条件 7 相符时为 ON	A274.06	A275.06	A320.06	A321.06
	与比较条件 8 相符时为 ON	A274.07	A275.07	A320.07	A321.07
比较动作中标志	执行比较动作中为 ON	A274.08	A275.08	A320.08	A321.08
溢出/下溢标志	线形模式下, 当前值为溢出/下溢时为 ON	A274.09	A275.09	A320.09	A321.09
计数器方向标志	0: 减法; 1: 加法	A274.10	A275.10	A320.10	A321.10
复位标志	用于当前值的软复位	A531.00	A531.01	A531.02	A531.03
高速计数器选通标志	为 ON 时, 禁止脉冲输入的计数, 保持计数器当前值	A531.08	A531.09	A531.10	A531.11

3. 高速计数器的控制指令

高速计数器控制指令在 CX-P 编程软件中属于脉冲控制类指令。表 5.10 是高速计数器控制指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。

表 5.10 高速计数器的控制指令(脉冲控制指令 1)

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位影响
比较表登录指令 CTBL/@CTBL	<div><div>CTBL(882)</div><div>P C TB</div></div> <div><div>@CTBL(882)</div><div>P C TB</div></div>	P 是端口定义 0000H~0003: 高速计数器输入 0~输入 3 C 是控制数据, 其含义为: 0000H: 登录一个目标值比较表, 并启动比较 0001H: 登录一个区域比较表, 并启动比较 0002H: 登录一个目标值比较表, 用 INI 启动比较 0003H: 登录一个区域比较表, 用 INI 启动比较 TB 是比较表开始通道 范围: CIO、W、H、T/C、A448~959、D、@/*D 等	执行条件为 ON 时, 根据 C 的内容, 登录一个目标值比较表或区域比较表; 根据 C 的内容, 决定启动比较的方式 下列情况之一时, ER 为 ON: ① 操作数超出指定范围 ② 高速计数器的设置有错误 ③ 比较表超出数据区或比较表的设置有错误
动作模式控制指令 INI/@INI	<div><div>INI(880)</div><div>P C S</div></div> <div><div>@INI(880)</div><div>P C S</div></div>	P 是端口定义 0000H~0003H: 脉冲输出 0~3 0010H~1013H: 高速计数器输入 0~3 0100H~0107H: 中断输入 0~7 1000H、1001H: PWM 输出 0、1 C 是控制数据, 其含义为: 0000H: 启动 CTBL 比较表 0001H: 停止 CTBL 比较表 0002H: 变更高速计数器、中断输入(计数模式)或脉冲输出当前值, 变更值设定在 S+1、S 中(S 的数值变更范围, 详见 CPlH CPU 单元编程手册) 0003H: 停止脉冲或 PWM 输出 在已经停止时, 则清除脉冲量设定 S 是存放变更数据设定值的低位通道, 只在 C=0002H 时有用, 范围为 CIO、W、H、T/C、A、D、@/*D 等	执行条件为 ON 时, 根据 C 的内容作如下操作之一: ① 启动或停止比较表的比较 ② 更新各种功能的当前值 ③ 停止脉冲输出 下列情况之一时, ER 为 ON: ① 操作数超出指定范围 ② 指令的设置错误 ③ 中断任务中执行了 INI 指令

(续表)

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位影响
当前值读出指令 PRV/@PRV	<div><div>PRV(881) P C D</div><div>@PRV(881) P C D</div></div>	<p>P 是端口定义, 同上面 INI 指令</p> <p>C 是控制数据, 其含义为:</p> <p>0000H: 读取当前值</p> <p>0001H: 读取状态</p> <p>0002H: 读取区域比较结果</p> <p>0003H: P=0000H~0003H 时, 读取脉冲输出 0~3 的频率。P=0010H 时, 读取高速计数器输入 0 的频率</p> <p>0013H: 频率为 10 ms 采样方式</p> <p>0023H: 为 100 ms 采样方式</p> <p>0033H: 为 1 s 采样方式</p> <p>D 是当前值保存低位 CH 号</p> <p>范围: CIO、W、H、T/C、A448~958、D、@/*D 等</p>	<p>执行条件为 ON 时:</p> <p>① 读取高速计数器或脉冲输出的当前值, 并保存到目的通道 D+1、D; 读取中断输入的当前值, 并保存到目的通道 D</p> <p>② 读取脉冲输出、PWM 输出, 或高速计数器输入状态信息, 并保存到目的通道 D (详见 CPIH CPU 单元编程手册)</p> <p>③ 读取区域比较结果, 并保存到目的通道 D。比较结果 1~8 分别存到 D.00~D.07</p> <p>④ 读取脉冲输出频率、高速计数器 0 的输入频率, 并保存到目的通道 D+1、D</p> <p>下列情况之一时, ER 为 ON:</p> <p>① 操作数超出指定范围</p> <p>② 操作数设置不匹配</p> <p>③ 在高速计数器 0 以外, 指定并读取输入频率</p>

下面举例说明利用高速计数器控制指令编写各种中断控制程序的方法。

**例5-3-3** 设计程序使高速计数器0 在线形模式下使用, 当前值达到 30 000 (BCD) (00007530 Hex) 时, 使中断任务 10 启动。

① 在“PLC 设定”对话框的“内置输入设置”选项卡中进行高速计数器 0 的设定, 见表 5.11。

表 5.11 内置输入设置

项 目	设 定 内 容
高速计数器 0	使用
计数模式	线形模式
循环最大计数	—
复位方式	软复位
输入(模式)设置	加/减输入

- ② 添加中断任务 10。
- ③ 将目标值比较表数据编制为 D10000~D10003, 见表 5.12。

表 5.12 目标值比较表数据

地址	设定值	内 容	
D10000	#0001	比较个数	1 个
D10001	#7530	目标值 1 数据 30000 的 Hex 值的低位 4 位	目标值 30000
D10002	#0000	目标值 1 数据 30000 的 Hex 值的高位 4 位	
D10003	#000A	目标值 1 位 15: 0 表明加法 位 0～07: A Hex 表明中断任务 No. 10	

④ 程序如图5.22 (a)所示, CTBL 指令的第 1 个操作数为 0000, 表示指定高速计数器 0 的输入; 第 2 个操作数为 0000, 表示登录一个目标值比较表并开始进行比较; 第 3 个操作数为 D10000, 是比较表的开始通道。目标值比较表中设了 1 个目标值。

执行条件 W0.00 为 ON 时, 开始高速计数器 0 的比较动作。高速计数器 0 的当前值达到 30 000 时, 则中断周期执行任务的处理, 进行中断任务 10 的处理。如中断任务 10 的处理结束, 则再次开始已中断的周期执行任务的处理, 动作过程如图5.22 (b)所示。

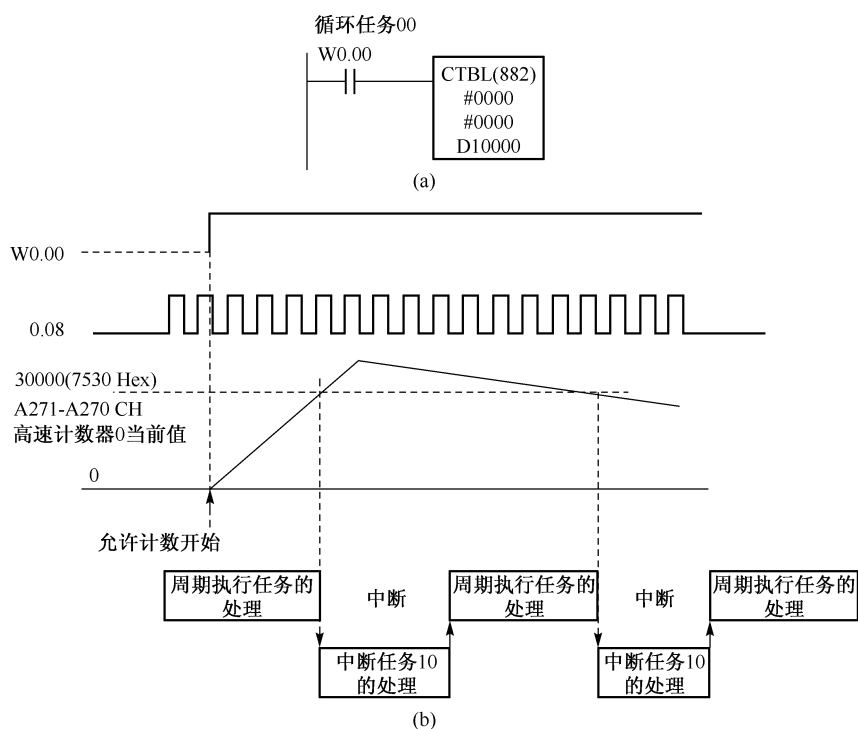


图 5.22 高速计数器目标值比较中断示例 1

**例 5-3-4** 图5.23(a)是采用高速计数器目标值比较中断的例子,图5.23(b)是目标值比较表的内容。编写高速计数器中断处理程序时,要用 CX-P 编程软件先添加高速计数器对应的中断任务,中断任务号可选择中断任务 0~255。

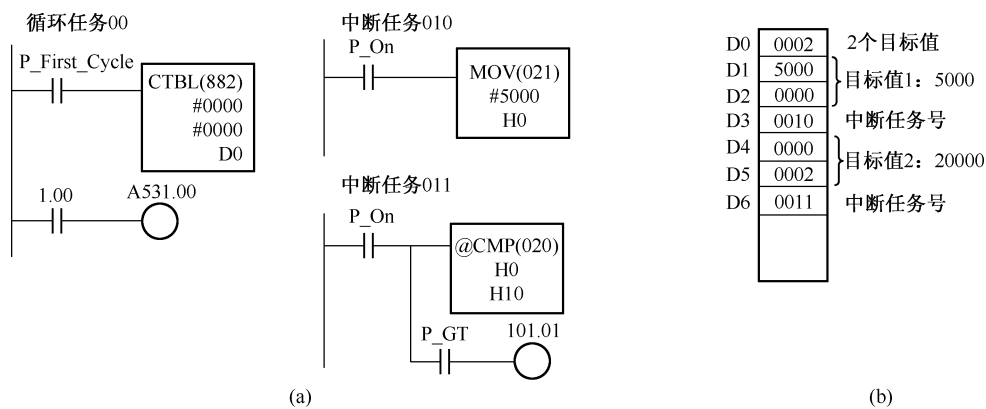


图 5.23 高速计数器目标值比较中断示例 2

图5.23(a)中, CTBL 指令的第 1 个操作数为 0000, 表示指定高速计数器 0 的输入; 第 2 个操作数为 0000, 表示登录一个目标值比较表并开始进行比较; D0 是比较表的开始通道。图 5.23(b)的目标值比较表中设了两个目标值。

图 5.23(a)中, 若高速计数器 0 的当前值(存于 A271、A270)等于目标值 1 时, 中断循环程序而执行 010 号中断任务, 把#5000 传送到 H0 中。中断任务 010 执行完毕返回断点处继续执行循环程序(本例没写其他循环程序)。若高速计数器的当前值等于目标值 2 时, 中断主程序而执行 011 号中断任务, 将 H0 与

H10 中的内容进行一次比较,若 H0 的内容大于 H10 时,101.01 为 ON。中断任务 011 执行完毕返回断点处继续执行循环程序。A531.00 为高速计数器 0 的软复位,若 1.00ON 且有 Z 信号时,高速计数器复位。

例 5-3-5  区域比较中断例子。设计程序使高速计数器 1 在循环模式下使用,当前值达到 25000~25 500(BCD) (000061A8~0000639C Hex)的范围时,使中断任务 12 启动。环形计数器的最大值设为 50000 (0000C350 Hex)。

① 在“PLC 设定”对话框的“内置输入设置”选项卡中进行高速计数器 1 的设定,见表 5.13。

表 5.13  内置输入设置

项    目	设  定  内  容
高速计数器 1	使用
计数模式	循环模式
循环最大计数	50 000
复位方式	软复位 (比较继续)
输入 (模式) 设置	加/减输入

② 添加中断任务 12。

③ 将区域比较表数据编制为 D20000 开始的表格,见表 5.14。

表 5.14  区域比较表数据

地 址	设 定 值	内 容	
D20000	#61A8	区域 1 下限值的低位 4 位	下限值 25 000
D20001	#0000	区域 1 下限值的高位 4 位	
D20002	#639C	区域 1 上限值的低位 4 位	上限值 25 500
D20003	#0000	区域 1 上限值的高位 4 位	
D20004	#000C	区域 1 中断任务 No.12 (C Hex)	
D20005～D20008	#0000	区域 1 的上限/下限数据 (因不使用, 无须设定)	区域 2 的设定区域
D20009	#FFFF	因不使用, 设为#FFFF	
⋮ ⋮ ⋮			
D20014 D20019 D20024 D20029 D20034	#FFFF	区域 3～7 的第 5 个字的数据(左侧所示)一定要设定#FFFF	
⋮ ⋮ ⋮			
D20035～D20038	#0000	区域 8 的上限/下限数据 (因不使用, 无须设定)	区域 8 的设定区域
D20039	#FFFF	因不使用, 设为#FFFF	

④ 在中断任务 12 中编制中断处理的程序。

⑤ 如图 5.24 (a) 所示,CTBL 指令的第 1 个操作数为 0001,表示指定高速计数器 1 的输入;第 2 个操作数为 0001,表示登录一个区域比较表并开始进行比较;第 3 个操作数 D20000 是比较表的开始通道。目标值比较表中设了 1 个目标值。

执行条件 W0.00 为 ON 时,开始高速计数器 1 的比较动作。高速计数器 1 的当前值达到 25000~25500,则中断周期执行任务的处理,进行中断任务的处理。如中断任务 12 的处理结束,则再次开始已中断的周期执行任务的处理,动作过程如图 5.24 (b) 所示。

例 5-3-6  图 5.25 是高速计数器区域比较中断的例子,图 5.25 (b) 是区域比较表的内容。

程序运行前要设置计数器的功能。本例采用高速计数器 0、加/减计数方式,复位方式是采用 Z 相和软件复位。

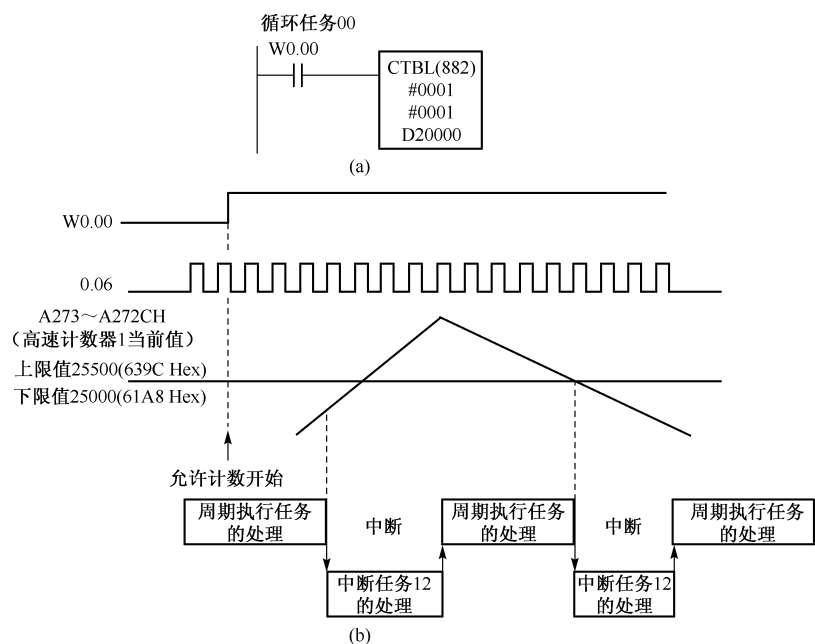


图 5.24 高速计数器区域比较中断的示例 1

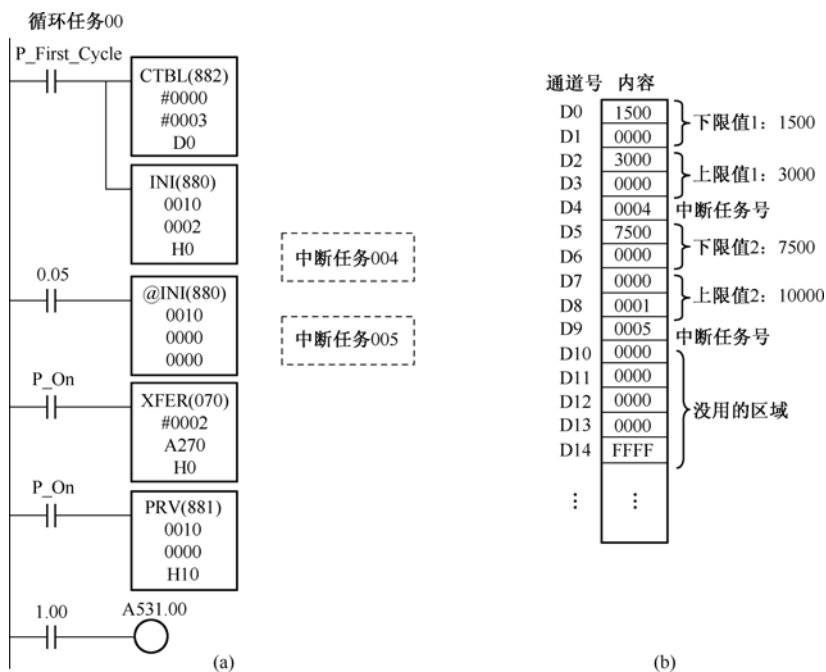


图 5.25 高速计数器区域比较中断的示例 2

图5.25(a)中 CTBL 指令的操作数 P 为 0000，表示指定高速计数器 0 的输入；C 为 0003，表示登录一个区域比较表，并用 INI 指令启动比较；D0 是区域比较表的开始通道。

图5.25(a)中用了两个 INI 指令，其中的非微分型 INI 指令执行的操作是，在 PLC 加电的第一个扫描周期中，将 H0 和 H1 两个通道的内容(PLC 断电前瞬时的高速计数器的当前值)传送到高速计数器 0 的当前值寄存器 A271、A270 中，以作为高速计数器 0 的新当前值。这样做的目的是，使 PLC 加电前、后高速计数器的当前值连续，这种做法在控制中有一定的实际意义。微分型 INI 指令用来启动比较。



在 0.05 由 OFF 变为 ON 时执行一次 INI 指令,使高速计数器的当前值开始与 CTBL 指令所登录的区域比较表进行比较,即 CTBL 指令所登录的区域比较表在 0.05 为 ON 时才开始启动比较。

图5.25 (b)的区域比较表是设在 D0~D39 这 40 个通道中,本例表中只设定了两个比较区域,因此其余 6 个区域中存放上、下限值的通道都置为#0000,存放中断任务号的通道都置为#FFFF。

本例的中断执行过程是,若高速计数器 0 的当前值落在区域 1 中时,中断循环程序,转去执 004 号中断任务,执行完毕返回断点处继续执行循环程序;若高速计数器的当前值落在区域 2 中时,中断执行循环程序,转去执行 005 号中断任务,执行完毕返回断点处继续执行循环程序。

图5.25 (a)中还使用了块传送指令 XFER (070),执行该指令是将高速计数器 0 的当前值寄存器 A271、A270 两个通道的内容传送到 H1、H0 中。这样做的目的是,一旦 PLC 掉电,高速计数器的当前值能被保存在 H1、H0 中,再加电时通过执行第一个 INI 指令,就可以把掉电前的当前值传送到高速计数器 0 的当前值通道 A271、A270 中,以作为高速计数器的新当前值,使 PLC 加电前、后高速计数器 0 的当前值连续。图 5.25 (a)中还使用当前值读出指令 PRV,目的是随时将 A271、A270 中的当前值读到 H11、H10 中去。

若 1.00 ON 且有 Z 信号,则高速计数器 0 复位。

例 5-3-7 通过脉冲输入的计数进行的尺寸检查,要求:

① 使用 X 型 AC 电源规格 CP1H 高速计数器 0 对工件顶端检测;计数值在 30000~30300 范围内时为合格,除此以外为不合格。

② 合格品的情况下,用中断将 OUT100.00 置于 ON,使 PL1 灯亮。不合格品的情况下,用中断将 OUT100.01 置于 ON,使 PL2 灯亮。

③ 中断程序在中断任务 10 中编制,采用 Z 相和软件复位方式复位。

按要求进行以下的工作。

① 输入/输出点分配,见表 5.15。

表 5.15  输入/输出点分配

输入/输出点			用    途
输入点	通道	位	
	0 CH	00	测量开始按钮(通用输入)
		01	测量物终端检测顶端通用输入(通用输入)
		03	测量物体顶端检测高速计数器 0(Z 相/复位)反映到 A531.00
		08	高速计数器 0(A 相输入)
		09	高速计数器 0(B 相输入)
输出点	100 CH	00	通用输出。PL1: 尺寸合格输出
		01	通用输出。PL2: 尺寸不合格输出

② 通过 PLC 系统设定,选“使用高速计数器 0”,高速计数器 0 的使用区域见表 5.16。

表 5.16  高速计数器 0 的使用区域

内    容		高速计数器 0
当前值保存区域	保存高位 4 位	A271 CH
	保存低位 4 位	A270 CH
区域比较一致标志	与比较条件 1 相符时为 ON	A274.00
比较动作中标志	执行比较动作中为 ON	A274.08
溢出/下溢标志	线形模式下,当前值为溢出/下溢时为 ON	A274.09
计数方向标志	0: 减法计数时  1: 加法计数时	A274.10
复位标志	用当前值的软复位	A531.00
高速计数器选通标志	选通标志为 1(ON)时,禁止进行脉冲输入的计数	A531.08

③ PLC 系统设定。“PLC 设定”的“内置输入设置”“高速计数器 0 的使用”的设定，见表 5.17。

表 5.17 高速计数器 0 设定

项 目	设 定 内 容
高速计数器 0	使用
计数模式	线形模式
循环最大计数	—
复位方式	Z 相和软件复位
输入(模式)设置	加法脉冲输入

④ 输入/输出的布线。输入布线如图5.26所示。输出布线如图5.27所示。

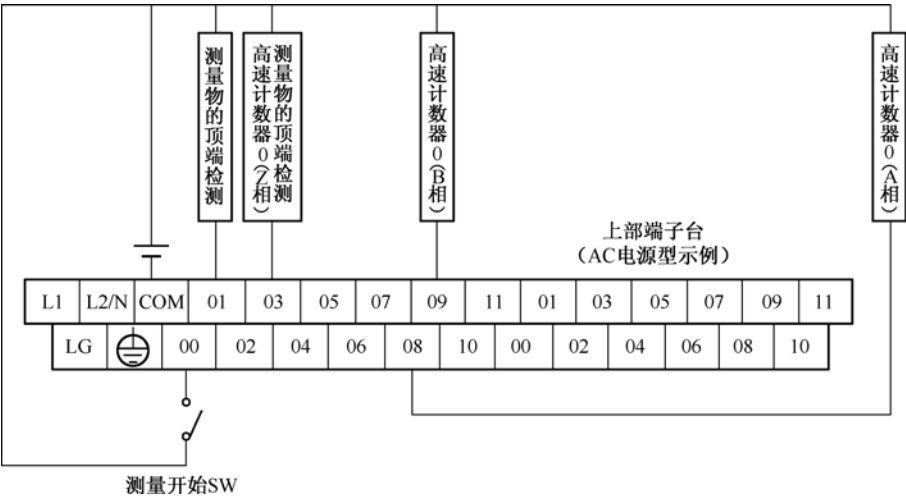


图 5.26 输入布线图

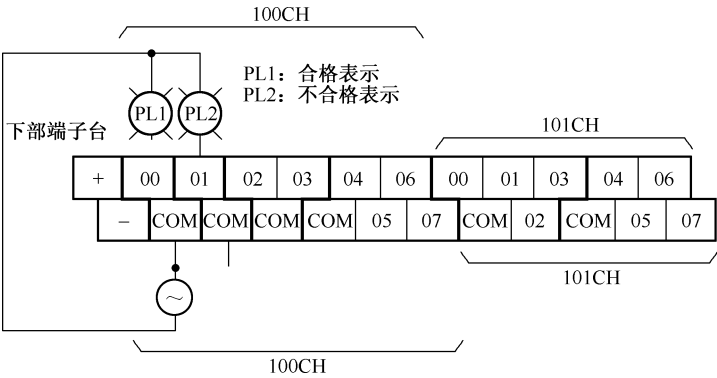


图 5.27 输出布线图

⑤ 将区域比较表数据编制为 D10000 开始的表格，见表 5.18。

表 5.18 区域比较表

地 址	设 定 值	内 容	
D10000	#7530	区域 1 下限值的低位 4 位	下限值 30000
D10001	#0000	区域 1 下限值的高位 4 位	
D10002	#765C	区域 1 上限值的低位 4 位	上限值 30300
D10003	#0000	区域 1 上限值的高位 4 位	
D10004	#000A	区域 1 中断任务 No.10 (A Hex)	

(续表)

地  址	设  定  值	内  容	
D10005~D10008	#0000	区域 2 的上限/下限数据 (因不使用, 无须设定)	区域 2 的设定区域
D10009	#FFFF	因不使用, 设为#FFFF	
∴ ∴ ∴			
D10014 D10019 D10024 D10029 D10034	#FFFF	区域 3~7 的第 5 个字的数据(左侧所示)一定要设定#FFFF	
∴ ∴ ∴			
D10035~D10038	#0000	区域 8 的上限/下限数据 (因不使用, 无须设定)	区域 8 的设定区域
D10039	#FFFF	因不使用, 设为#FFFF	

⑥ 梯形图程序的编制

周期执行任务的程序如图5.28所示。通过 CTBL 指令，设定高速计数器 1 的比较动作、中断任务 10 的启动。

中断任务 10 的程序如图5.29所示。

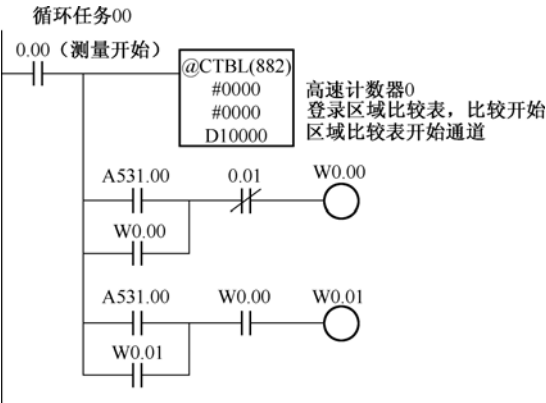


图 5.28 循环程序图

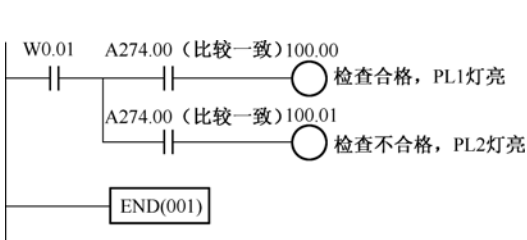


图 5.29 中断程序图

综上所述，高速计数器具有高速计数和中断功能，现归纳如下。

- ① 使用高速计数器前必须进行设定，在 PLC 的 CX-P 编程软件的工程目录下，双击“设置”，弹出“PLC 设定”对话框，在“内置输入设置”选项卡进行设置，以确定高速计数器的使用/不使用、计数模式、输入模式和复位方式等。
- ② 使用高速计数器时，其当前值存储通道 A271 和 A270(高速计数器 0)、A273 和 A272(高速计数器 1)、A317 和 A316(高速计数器 2)、A319 和 A318(高速计数器 3)已经被占用时，不能再作他用。
- ③ 使用高速计数时相应的输入点被占用，不能再作他用：  
高速计数器 0 占用输入端 0.03、0.08、0.09；  
高速计数器 1 占用输入端 0.02、0.06、0.07；  
高速计数器 2 占用输入端 0.01、0.04、0.05；  
高速计数器 3 占用输入端 1.00、0.10、0.11。
- ④ 高速计数器有计数功能。递增模式计数时，计数脉冲可以是外部输入的信号或旋转编码器输出的单脉脉冲。加/减模式计数时可用旋转编码器的输出脉冲作为计数脉冲，旋转编码器正转时为递增计数，反转时为递减计数。

- ⑤ 高速计数器具有中断功能。在使用其中断功能时，要用 CTBL 指令登录一个目标值比较表或区域比较表。所登录的比较表可以立即启动比较，也可以用 INI 启动比较。
- ⑥ 高速计数器的中断任务 0~255 与普通程序的编写规则相同。

5.3.4 脉冲输出控制指令

CP1 系列 PLC 可从 CPU 单元内置输出中发出固定和可变占空比两种脉冲输出信号，并通过脉冲输入的伺服电动机驱动器进行定位/速度控制。下面主要以 CP1H PLC 为例介绍脉冲输出控制指令。其他 CP1 系列 PLC 的脉冲输出功能简单加以说明。

1. 脉冲输出端分配

CP1H 系列晶体管输出型 PLC 有 4 对固定脉冲输出端子，两对可变占空比脉冲输出端子。CP1L/CP1E 有两对固定脉冲输出端子，这些 PLC 端子分配见表 5.19，其中 CW 表示顺时针方向，CCW 表示逆时针方向。

表 5.19 CP1 系列 PLC 脉冲输出端分配

输出点		固定脉冲输出种类	固定脉冲输出方式		可变占空比脉冲输出	输出频率范围		
						CP1H	CP1L	CP1E
通道	位		CW/CCW	脉冲+方向	—			
100CH	00	脉冲输出 0	CW(顺时针)	脉冲	—	X/XA 型 1 Hz~100 kHz Y 型 1 Hz~1 MHz	1 Hz~100 kHz	1 Hz~100 kHz (限 N/NA 型)
	01	脉冲输出 0	CCW(逆时针)	方向	—			
	02	脉冲输出 1	CW	脉冲	—			
	03	脉冲输出 1	CCW	方向	—			
	04	脉冲输出 2	CW	脉冲	—	X/XA 型 1 Hz~100 kHz Y 型 1 Hz~100 kHz	无	无
	05	脉冲输出 2	CCW	方向	—			
	06	脉冲输出 3	CW	脉冲	—			
	07	脉冲输出 3	CCW	方向	—			
100CH	01	—	—	—	PWM 输出 0	无	有	有
	03	—	—	—	PWM 输出 1	无	有	无
101CH	00	—	—	—	PWM 输出 0	有	无	无
	01	—	—	—	PWM 输出 1	有	无	无

2. 脉冲输出方式

脉冲输出有两种输出方式：CW/CCW 输出和脉冲+方向输出。

- ① 选择 CW/CCW 输出时，以脉冲输出 0 为例，则 100.00 输出 CW 方向的脉冲，100.01 输出 CCW 方向的脉冲。
- ② 选择脉冲+方向输出时，以脉冲输出 0 为例，当 100.01 为 ON 时，100.00 输出顺时针方向脉冲；当 100.01 为 OFF 时，100.00 输出逆时针方向脉冲。

3. 脉冲输出模式

脉冲输出可以设置成连续模式或独立模式。在设置成连续模式时，要用指令来控制脉冲的停止(INI 指令)；在设置成独立模式时，须由 PULS 指令设定脉冲输出量，如图 5.30 所示。

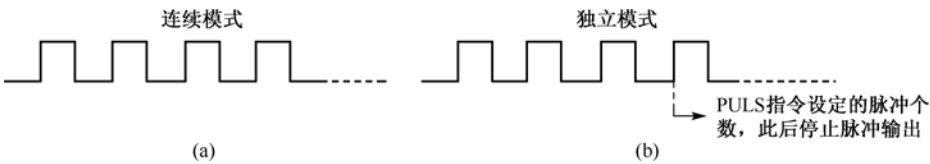


图 5.30 脉冲输出的两种模式

4. 脉冲输出控制类指令

表5.20 是脉冲输出控制类指令的名称、梯形图符号、操作数的含义及范围、指令功能及执行指令对标志位的影响。

表 5.20 脉冲输出控制类指令 (脉冲控制指令 2)

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位影响
设置脉冲指令 PULS/@PULS	<div><div><div>PULS(886)</div><div>P C N</div></div><div><div>@PULS(886)</div><div>P C N</div></div></div>	<p>P 是端口定义 0000H~0003H: 脉冲输出 0~3</p> <p>C 是控制数据 0000H: 设定相对脉冲 0001H: 设定绝对脉冲</p> <p>N 是存放设置脉冲个数的首通道 相对脉冲时, 数值范围: 0~2147483647 (0~7FFFFFFFHex) 绝对脉冲时, 数值范围: -2147483648~2147483647 (80000000~7FFFFFFFHex)</p> <p>继电器区范围: CIO、W、H、T/C、A、D、@/*D、#等</p>	<p>执行条件为 ON 时, 设定 SPED 指令独立模式的脉冲输出个数</p> <p>在下列情况之一时, ER 为 ON:</p> <ul style="list-style-type: none"><li>① 超出 P、C、N 的范围</li><li>② 已使用本指令的正在进行脉冲输出的端口, 再使用本指令时</li><li>③ 在用周期执行任务执行控制脉冲输出指令中, 需要中断, 在中断任务内执行本指令时</li></ul>
脉冲输出指令 SPED/@SPED	<div><div><div>SPED(885)</div><div>P M F</div></div><div><div>@SPED(885)</div><div>P M F</div></div></div>	<p>P 是端口定义 0000H~0003H: 脉冲输出 0~3</p> <p>M 是脉冲输出模式控制字</p> <div><div>固定0</div><div><div>模式设定 0: 连续, 1: 独立</div><div>方向设定 0: CW, 1: CCW</div><div>脉冲输出方式 0: CC/CCW, 1: 脉冲+方向输出 (同时使用脉冲输出0、1进行动作时, 须采用相同方式)</div></div></div> <p>F 是脉冲输出频率首字, 0~100 kHz 范围: CIO、W、H、T/C、A、D、@/*D、#等</p>	<p>执行条件为 ON 时, 由 P 指定的端口, 通过 M 指定的模式和 F 指定的频率输出脉冲</p> <p>注: 当模式设定为连续模式时, 用 INI 指令停止脉冲的输出; 当模式设定为独立模式时, 用 PULS 指令设定脉冲输出的脉冲个数</p> <p>在下列情况之一时, ER 为 ON:</p> <ul style="list-style-type: none"><li>① 超出 P、M、F 的范围时</li><li>② 对在 ORG、PL2 指令中, 正在进行脉冲输出的端口使用本指令时</li><li>③ 对在 SPED、ACC 指令中, 正在进行脉冲输出的端口使用本指令改变独立/连续模式时</li><li>④ 在周期执行任务执行控制脉冲输出指令中, 需要中断, 在中断任务内执行本指令时</li><li>⑤ 在未确定点时, 由独立模式□绝对脉冲指定来执行本指令时</li></ul>
频率加速度控制指令 ACC/@ACC	<div><div><div>ACC(888)</div><div>P M F</div></div><div><div>@ACC(888)</div><div>P M F</div></div></div>	<p>P 是端口定义 0000H~0003H: 脉冲输出 0~3</p> <p>M 是脉冲输出模式控制字</p> <div><div>固定0</div><div><div>模式设定 0: 连续, 1: 独立</div><div>方向设定 0: CW, 1: CCW</div><div>脉冲输出方式 0: CC/CCW, 1: 脉冲+方向输出 (同时使用脉冲输出0、1进行动作时, 须采用相同方式)</div></div></div> <p>F 是加减速比率通道 1~65535Hz/4ms (1~FFFFHex) F+1 是目标频率低位通道 F+2 是目标频率高位通道 0~100 kHz (0~000186A0Hex) 相对脉冲时, 范围: 0~2147483647 绝对脉冲时, 范围: -2147483648~2147483647 范围: CIO、W、H、T/C、A、D、@/*D 等</p>	<p>执行条件为 ON 时, 由 P 指定的端口, 通过 M 指定的模式和 F 指定加减速率和 F+2、F+1 指定的目标频率输出脉冲</p> <p>注: ① 独立模式时, 要与 PULS 指令配合使用。 ② 加速率=减速率</p> <p>在下列情况之一时, ER 为 ON:</p> <ul style="list-style-type: none"><li>① 超出 P、M、F 的范围时</li><li>② 对在 ORG 指令中, 正在进行脉冲输出的端口使用本指令时</li><li>③ 对在 SPED、ACC、PL2 指令中, 正在进行脉冲输出的端口使用本指令改变独立/连续模式时</li><li>④ 在周期执行任务执行控制脉冲输出指令中, 需要中断, 在中断任务内执行本指令时</li><li>⑤ 在未确定原点时, 由独立模式□绝对脉冲指定来执行本指令时</li></ul>

(续表)

名 称	梯形图符号	操作数的含义及范围	指令功能及执行指令对标志位影响
脉冲输出 定位指令 PLS2/@PLS2	<div><div>PLS2(887) P M F1 F2</div><div>@PLS2(887) P M F1 F2</div></div>	<p>P 是端口定义 0000H~0003H: 脉冲输出 0~3 M 是脉冲输出模式控制字</p> <div><div><div>固定0</div><div>模式设定 0: 相对脉冲, 1: 绝对脉冲</div><div>方向设定 0: CW, 1: CCW</div><div>脉冲输出方式 0: CC/CCW, 1: 脉冲+方向输出 (同时使用脉冲输出0、1进行动作时, 须采用相同方式)</div></div></div> <p>F1 是加速比率通道 F1+1 是减速比率通道 1~65535 Hz/4ms (1~FFFFHex) F1+2 是目标频率低位通道 F1+3 是目标频率高位通道 0~100 kHz (0~000186A0Hex) F1+4 是脉冲输出设定值低位通道 F1+5 是脉冲输出设定值高位通道 相对脉冲时, 范围: 0~2147483647 (0~7FFFFFFFHex) 绝对脉冲时, 范围: -2147483648~2147483647 (80000000~7FFFFFFFHex) F2 是启动频率低位通道 F2+1 是启动频率高位通道 0~100 kHz (0~000186A0Hex) 范围: CIO、W、H、T/C、A、D、@/*D 等</p>	<p>执行条件为 ON 时, 由 P 指定的端口, 通过 M 指定的模式和 F1+5~F1 指定脉冲输出频率特性, F2+1、F2 指定的启动频率, 输出脉冲, 实现定位</p> <p>在下列情况之一时, ER 为 ON:</p> <p>① 超出 P、M、F1、F2 的范围 ② 对在 SPED、ORG 指令中, 正在进行脉冲输出的端口使用本指令 ③ 在周期执行任务执行控制脉冲输出指令中, 需要中断, 在中断任务内执行本指令时④ 在未确定原点时, 以绝对脉冲模式来执行本指令</p>
原点搜索指令 ORG/@ORG	<div><div>ORG(889) P C</div><div>@ORG(889) P C</div></div>	<p>P 是端口定义 0000H~0003H: 脉冲输出 0~3 C 是脉冲输出模式控制字</p> <div><div><div>固定0</div><div>固定0</div><div>脉冲输出方式 0: CC/CCW, 1: 脉冲+方向输出 (脉冲输出0、1须采用相同方式)</div><div>模式设定 0: 原点搜索, 1: 原点复位</div></div></div>	<p>从由 P 指定的端口中, 用由 C 指定的方式输出脉冲, 进行原点搜索或原点复位</p> <p>① 原点搜索: 根据事先设定的方式, 通过输出脉冲, 来实际启动电动机, 使用原点附近的输入信号、原点输入信号来确定原点 ② 原点复位: 从当前的位置向确定的原点位置移动</p> <p>在下列情况之一时, ER 为 ON:</p> <p>① 超出 P、C 的范围时 ② 对在 SPED、ACC、PL2 指令中, 正在进行脉冲输出的端口使用本指令时 ③ 在周期执行任务执行控制脉冲输出指令中, 需要中断, 在中断任务内执行本指令时 ④ 未确定原点的状态下指定原点复位时 ⑤ 其他设置错误时</p>
PWM 输出指令 (可变占空比脉冲输出) PWM/@PWM	<div><div>PWM(891) P F C</div><div>@PWM(891) P F C</div></div>	<p>P 是端口定义 1000H: PWM 脉冲输出 0 1001H: PWM 脉冲输出 1 F 频率指定 0001H~FFFFH: 0.1~6553.5Hz 精度 0.1% C 占空比指定 0000H~03E8H: 0.0~100.0%精度 0.1% F、C 的范围: CIO、W、H、T/C、A、D、@/*D、DR、#等</p>	<p>执行条件为 ON 时, 从指定端口中输出指定占空比的脉冲</p> <p>在下列情况之一时, ER 为 ON</p> <p>① 超出 P、F、C 的范围时 ② 对在 ORG 指令中, 正在进行脉冲输出的端口使用本指令时 ③ 在周期执行任务执行控制脉冲输出指令中, 需要中断, 在中断任务内执行本指令时</p>

5. 脉冲输出的状态区

脉冲输出的状态区是指脉冲输出的各种信息存放的指定的数据区，查看它们可以了解脉冲输出的工作情况。脉冲输出的状态区见表 5.21。

表 5.21  高速计数器的状态区

内    容		脉冲输出 0	脉冲输出 1	脉冲输出 2	脉冲输出 3
当前值保存区域	高位 4 位	A277CH	A279CH	A323 CH	A325CH
	低位 4 位	A276CH	A278CH	A322CH	A324 CH
脉冲输出复位标志：清除脉冲输出当前值	0：不清除 1：清除	A540.00	A541.00	A542.00	A543.00
CW 界限输入信号：原点搜索中使用的 CW 界限输入信号	来自外部的输入为 ON 时，为 ON	A540.08	A541.08	A542.08	A543.09
CCW 临界输入信号：原点搜索中使用的 CCW 界限输入信号	来自外部的输入为 ON 时，为 ON	A540.09	A541.09	A542.09	A543.09
定位完成信号：原点搜索中使用的定位完成信号	来自外部的输入为 ON 时，为 ON	A540.10	A541.10	A542.10	A543.10
脉冲输出状态标志：通过 ACC/PLS2 指令使脉冲输出中输出频率发生阶段性变化，并在加减速中为 ON	0：恒速中 1：加减速中	A280.00	A281.00	A326.00	A327.00
溢出/下溢标志：计数值为溢出或下溢时，为 ON	0：正常 1：发生中	A280.01	A281.01	A326.01	A327.01
脉冲输出量设定标志：通过 PLUS 指令，设定脉冲量时，为 ON	0：无设定 1：有设定	A280.02	A281.02	A326.02	A327.02
脉冲输出完成标志：通过 PULS/PLS2 指令设定的脉冲量结束输出时，为 ON	0：输出未完成 1：输出完成	A280.03	A281.03	A326.03	A327.03
脉冲输出中标志：脉冲输出中时，为 ON	0：停止 1：输出中	A280.04	A281.04	A326.04	A327.04
无原点标志：原点未确定时，为 ON	0：确认 1：未确认	A280.05	A281.05	A326.05	A327.05
原点停止标志：脉冲输出的当前值与原点 (0)一致时，为 ON	0：未在原点停止 1：在原点处停止	A280.06	A281.06	A326.06	A327.06
脉冲输出停止异常标志：原点搜索功能下，在脉冲输出中发生异常时为 ON	0：无异常 1：异常发生	A280.07	A281.07	A326.07	A327.07
停止异常代码：脉冲输出发生停止异常时，保存该异常代码		A444	A445	A438	A439

6. 指令应用示例

例 5-3-8 图 5.31 是连续模式脉冲输出的例子。图中，SPED 指令的操作数 P 为 0000，表示脉冲是从 100.00 和 100.01 输出，M 为 0000，表示为连续模式，F 为 0150，表示输出脉冲的频率是 150 Hz（见表 5.20）。INI 指令的操作数 P 为 0000，指从端口 100.00 和 100.01 输出。C 为 0003 表示当其执行条件为 ON 时停止脉冲输出（见表 5.10），该图的控制功能如下。

当执行条件 0.00 由 OFF 变为 ON 时，执行@SPED 指令启动脉冲输出，从 100.00 和 100.01 输出 150 Hz 的连续脉冲信号。当执行条件 0.01 由 OFF 变为 ON 时，执行@INI 指令停止脉冲输出。

例 5-3-9 图 5.32 是独立模式脉冲输出的例子。图中指令 PULS 的操作数表示设置的脉冲个数存放在 D0 中。指令 SPED 的操作数表示脉冲是从 100.00 和 100.01 输出，独立模式，输出脉冲的频率是 50 Hz。

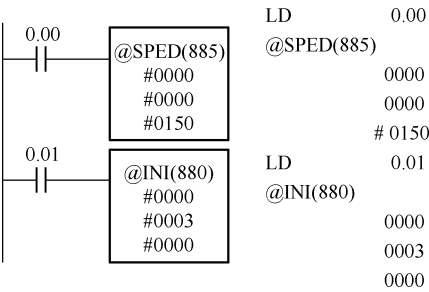


图 5.31 连续模式脉冲输出的例子

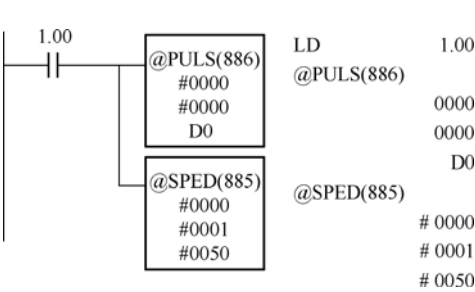


图 5.32 独立模式脉冲输出的例子

对图5.32来说，当脉冲输出指令的执行条件 1.00 由 OFF 变为 ON 时，执行@PULS 指令设置输出脉冲的个数(D0 的内容)，执行@SPED 指令启动脉冲输出，从 100.00 和 100.01 输出 50 Hz 的脉冲信号。当输出脉冲达到设定的脉冲个数时，自动停止脉冲输出。

在使用脉冲输出指令时，要注意以下几点：

- ① 正在输出脉冲时，不能用 PULS 指令改变输出脉冲的个数。
- ② 独立模式时，当达到指定脉冲数时停止脉冲输出；在连续输出模式时，当 SPED 指令的 F 设为 0000，或将 INI 指令的 C 设为 003，都可以使脉冲输出停止。

**例 5-3-10** @SPED、@INI 指令示例如图5.33所示。0.00 由 OFF 变为 ON 时，通过@SPED 指令，采用连续模式，开始从脉冲输出 0 中输出 500 Hz 的脉冲。0.01 由 OFF 变为 ON 时，通过@INI 指令停止脉冲输出。

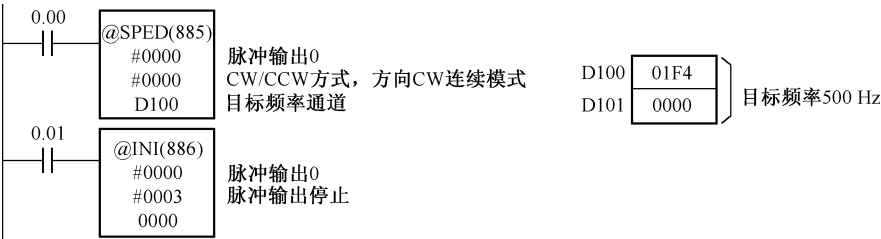


图 5.33 @SPED、@INI 指令示例

**例 5-3-11** @CTBL 指令示例如图5.34所示。@CTBL 指令的第 1 个操作数为#0000，表示高速计数器输入 0；第 2 个操作数为#0000，表示登录目标值比较表并开始比较，第 3 个操作数为 D100，表示目标值比较表开始通道为 D100。

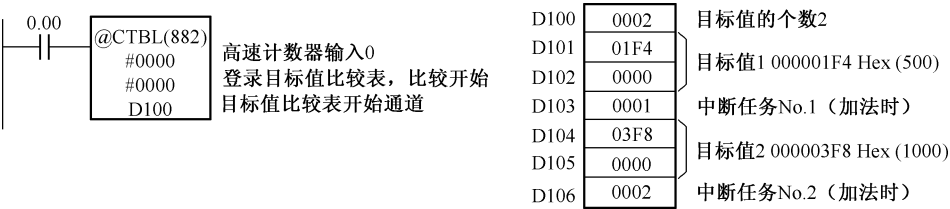


图 5.34 @CTBL 指令示例

0.00 由 OFF 变为 ON 时，由@CTBL 指令对高速计数器输入 0 进行目标值比较表的登录和比较。高速计数器当前值在加法方向上进行计数，当到达 500 时，由于和目标值 1 一致，因此执行中断任务 No.1。接着继续进行加法计数，当到达 1000 时，由于和目标值 2 一致，因此执行中断任务 No.2。

**例 5-3-12** @CTBL、@PRV 指令示例如图5.35所示，是输入中断(直接模式)的例子。



设定：① 在 PLC 的 CX-P 编程软件的工程窗口目录下，双击“设置”，弹出“PLC 设定”对话框，在“内置输入设置”选项卡中，中断输入“IN0”选为“中断”。

② 通过 CX-P 添加中断任务 140 的新程序，编制中断处理用的程序。

③ 通过 CX-P 编制程序，如图 5.35 所示。

图 5.35 中 @CTBL 指令的第 1 个操作数为 #0000，表示高速计数器输入 0；第 2 个操作数为 #0001，表示登录区域比较表并开始比较；第 3 个操作数为 D100，表示区域比较表开始通道为 D100。@PRV 指令的第 1 个操作数为 #0010，表示高速计数器输入 0；第 2 个操作数为 #0002，表示读取区域比较结果；第 3 个操作数为 100，表示输出到 100 通道中。

0.00 由 OFF 变为 ON 时，通过 @CTBL 指令，将区域比较表登录到高速计数输入 0 中，开始进行比较。0.01 由 OFF 变为 ON 时，通过 @PRV 指令，将该时的区域比较结果输出到 100 通道中。

例 5-3-13 @PRV 指令例子如图 5.36 所示。第 1 个操作数为 #0010，表示高速计数器输入 0；第 2 个操作数为 #0003，表示读频率；第 3 个操作数为 D200，表示数输出到 D201、D200 中。



图 5.35 @CTBL、@PRV 指令示例



图 5.36 @PRV 指令示例

0.01 为 ON 时，通过 @PRV 指令，读取输入到高速计数输入 0 中的脉冲频率，按 16 进制数格式输出到 D201、D200 中。

例 5-3-14 @PULS、@SPED 指令例子如图 5.37 所示。@PULS 指令的第 1 个操作数为 #0000，表示脉冲输出 0；第 2 个操作数为 #0000，表示相对脉冲指定模式；第 3 个操作数为 D100，表示输出脉冲量低位通道。@SPED 指令的第 1 个操作数为 #0000，表示脉冲输出 0；第 2 个操作数为 #0001，表示 CW/CCW 方式、CW 方向、独立模式；第 3 个操作数为 D110，表示目标频率低位通道。

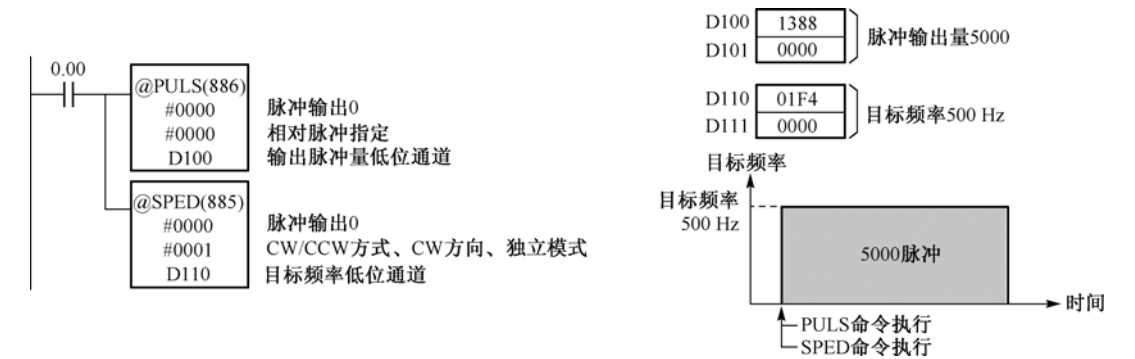


图 5.37 @PULS、@SPED 指令应用例子

0.00 由 OFF 变为 ON 时，通过 @PULS 指令由相对脉冲指定将脉冲输出 0 的脉冲输出量设定为 5000 个脉冲。同时通过 @SPED 指令由 CW/CCW 方式、CW 方向、独立模式开始输出目标频率 500 Hz 的脉冲，脉冲输出量设定为 5000 个。

例 5-3-15 @PLS2 指令例子如图 5.38 所示。@PLS2 指令的第 1 个操作数为 #0000，表示脉冲输

出 0；第 2 个操作数为#0000，表示 CW/CCW 方式、CW 方向、相对脉冲指定模式；第 3 个操作数为 D100，表示频率特性设定表低位通道；第 4 个操作数为 D110，表示启动频率低位通道。

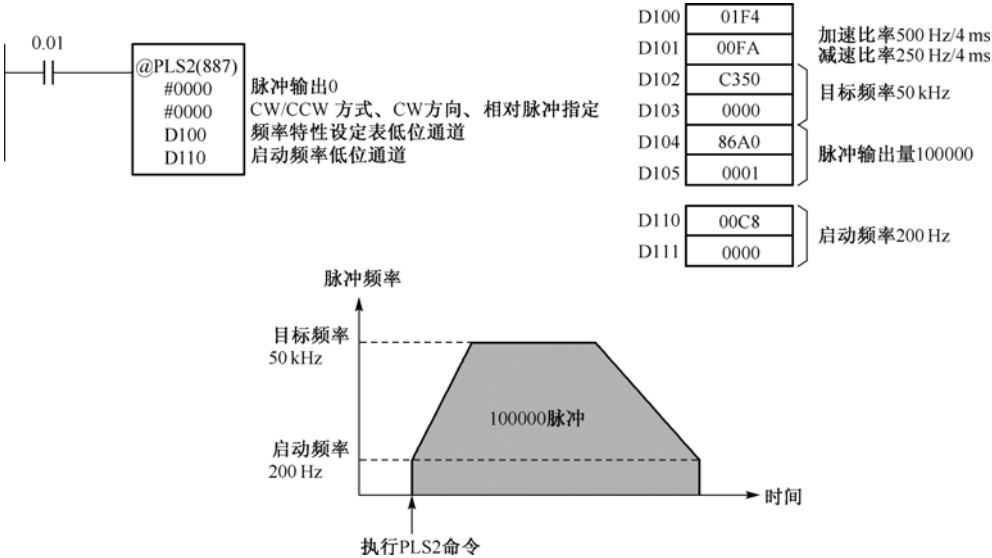


图 5.38 @PLS2 指令应用例子

0.01 由 OFF 变为 ON 时,通过@PLS2 指令由相对脉冲指定从脉冲输出 0 开始输出 100000 个脉冲。从启动频率的 200 Hz 开始,以 500 Hz/4 ms 的加速比率,加速到目标频率 50 kHz 为止。之后从减速点,以 250 Hz/4 ms 的减速比率进行减速,当减速到启动频率的 200 Hz 时,停止脉冲输出。

例 5-3-16 @ACC 指令例子如图5.39所示。第 1 个@ACC 指令的第 1 个操作数为#0000，表示脉冲输出 0；第 2 个操作数为#0000，表示 CW/CCW 方式、CW 方向、连续模式；第 3 个操作数为 D100，表示频率特性设定表低位通道。第 2 个@ACC 指令具有类似的含义。

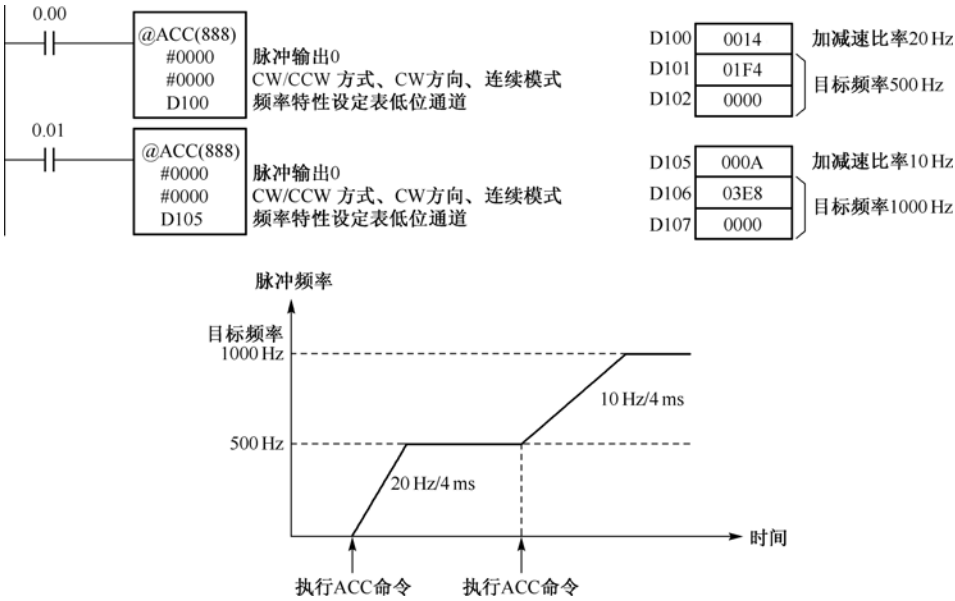


图 5.39 @ACC 指令应用例子

0.00 由 OFF 变为 ON 时, 通过@ACC 指令从脉冲输出 0 的端口, 用 CW/CCW 方式、CW 方向、连续模式开始进行加减速比率 20Hz、目标频率 500 Hz 的脉冲输出。之后 0.01 由 OFF 变为 ON 时, 再一次通过@ACC 指令变更为加减速比率 10 Hz、目标频率 1000 Hz。

**例 5-3-17 原点返回指令 ORG (889)。**返回原点指令以两种方式到达原点: 原点搜索和原点复位。两种方式的根本区别在于, 是否有依赖外部的原点附近的输入信号和原点输入信号来确定原点。

① 原点搜索。根据事先设定的方式, 通过输出脉冲, 来实际启动电动机, 使用原点附近的输入信号和原点输入信号来确定原点。工作过程如图5.40所示。

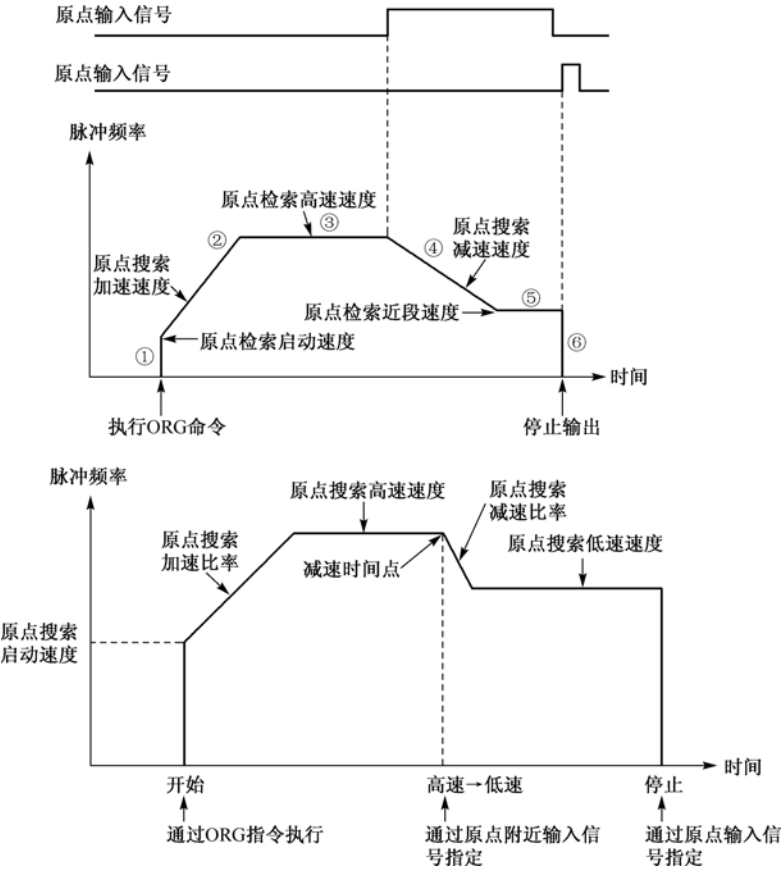


图 5.40 原点搜索方式工作过程

按指定的输出方式输出原点搜索启动速度的脉冲, 见图5.40中①段。随着原点搜索加速比率加速到原点搜索高速速度, 见图5.40中②段。在原点附近输入信号为 ON 之前, 保持等速, 见图5.40中③段。原点附近输入信号为 ON 时, 根据原点搜索减速比率, 一直减速到原点搜索附近速度, 见图5.40中④段。在原点输入信号为 ON 之前, 保持等速, 见图5.40中⑤段。原点输入信号为 ON 时, 停止脉冲输出, 见图5.40中⑥段。原点搜索完成时, 偏差计数重置输出为 ON。

② 原点复位。从当前的位置向确定的原点位置移动, 工作过程如图5.41所示。  
按指定的输出方式开始原点复位启动速度的脉冲输出, 见图5.41中①段。根据原点复位加速比率, 到原点复位目标速度为止进行加速, 见图5.41中②段。以等速进行动作, 见图5.41中③段。当到了从到原点位置为止的脉冲输出量和原点复位减速比率中计算得到的减速点时, 开始减速, 见图5.41中④段。当减速到原点复位启动速度时(到达原点), 停止脉冲输出, 见图5.41中⑤段。

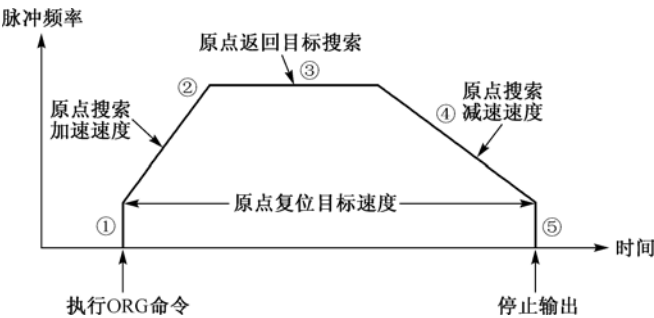


图 5.41 原点复位方式工作过程

@ORG 指令示例如图5.42所示，例子相对于 PLC 系统的设定值见表 5.22。启动速度为 100 pps（脉冲数/秒），目标速度为 200 pps，加速比率和减速比率为 50 Hz/4 ms。指令的第 1 个操作数为#0000，表示选择脉冲输出 0；指令的第 2 个操作数为#1000，表示用 CW/CCW 方式输出脉冲，进行原点复位。0.00 由 OFF 变为 ON 时，执行@ORG 指令，得到如图5.42 (b)所示的执行结果。

表 5.22 ORG 指令的设定值

名 称	PLC 系统设定值
脉冲输出 0 原点搜索/原点复位启动速度	00000064 Hex: 100 pps (脉冲数/秒)
脉冲输出 0 原点复位目标速度	000000C8 Hex: 200 pps
脉冲输出 0 原点复位加速比率	0032 Hex: 50 Hz/4 ms
脉冲输出 0 原点复位减速比率	0032 Hex: 50 Hz/4 ms

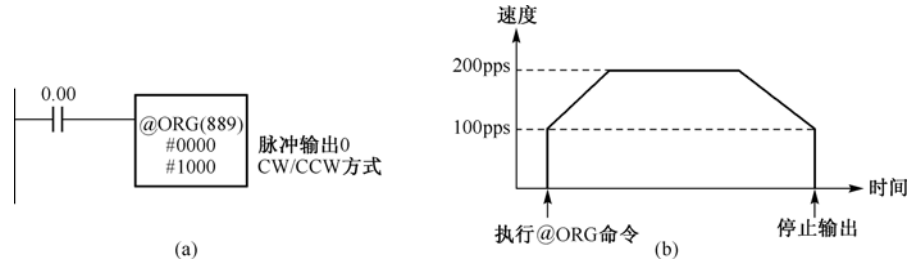


图 5.42 @ORG 指令示例

例 5-3-18 @PWM 指令示例如图5.43所示。第 1 个 PWM 指令的第 1 个操作数为#0000，表示脉冲输出 0；第 2 个操作数为#07D0，表示频率为 200 Hz；第 3 个操作数为#0032，表示占空比为 50%。第 2 个@PWM 指令具有类似的含义，两个@PWM 指令连用，目的是为了改变输出脉冲的占空比。

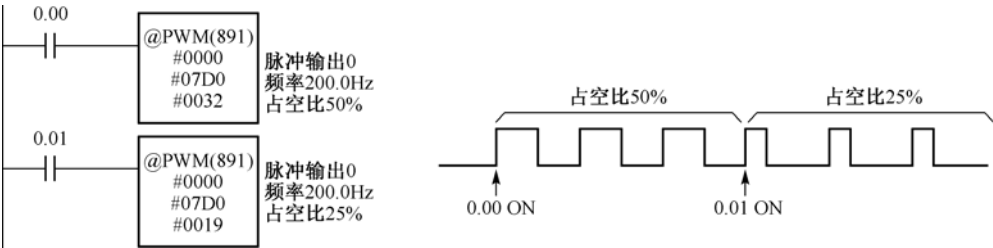


图 5.43 @PWM 指令示例

0.00 由 OFF 变为 ON 时，通过@PWM 指令对于脉冲输出 0 来说由频率 200 Hz、占空比 50%来开始脉冲的输出。0.01 由 OFF 变为 ON 时，占空比变更为 25%。

5.3.5  间隔定时器中断

间隔定时器中断(简称定时器中断)通过 CPU 单元的内置定时器，按照一定的时间间隔执行中断任务的处理。时间间隔的单位时间可从 10 ms、1 ms、0.1 ms 中选择。可设定的时间间隔为 0.5～99990 ms。中断任务 2 被固定分配为间隔定时器中断。间隔定时器中断不占用 CPlH 的输入点。

间隔定时器中断的使用步骤如下。

① 通过 CX-P 的 PLC 系统设定，设定定时中断处理单位时间。

在 CX-P 编程软件的工程工作区，双击项目“设置”，弹出“PLC 设定”对话框，在“时序”选项卡中，最下方的选项为“定时中断间隔”，如图5.44所示。这里选择的是定时中断的设定时间的单位。定时中断的执行周期为此处的设定时间单位与 MASK 指令的时间设定值的乘积。

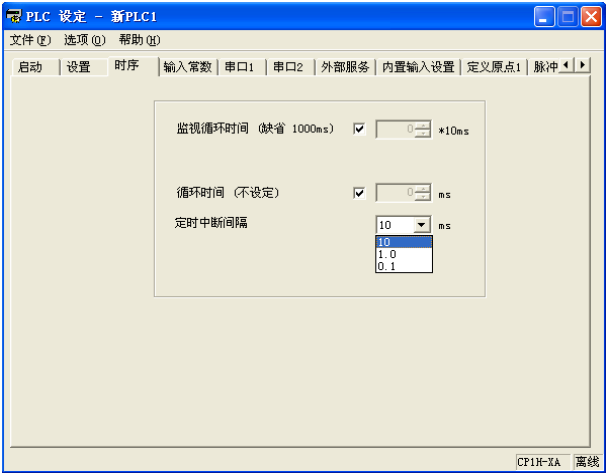


图 5.44 “PLC 设定”对话框

② 在中断任务 2(定时中断)中编制中断处理的程序。

在 CX-P 编程软件的工程工作区，右击项目树的“新 PLC1[CPlH]”，在其快捷菜单中选择“插入程序(I)”命令，则在树的下方出现“新程序 2(未指定)”，右击“新程序 2(未指定)”，在其快捷菜单中选择“属性”命令，弹出“程序属性”对话框，如图5.45所示，在“通用”选项卡中选择中断任务编号，例如选择“中断任务02(间隔定时器 0)”，即选择间隔定时器任务 0。

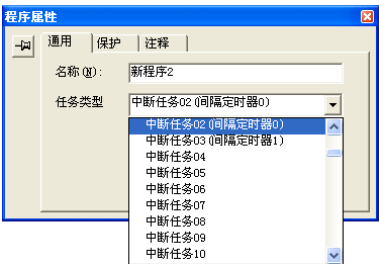


图 5.45 “程序属性”对话框

③ 通过 MSKS 指令设定定时中断处理时间，见表 5.23。

表 5.23  定时中断处理时间的设定

MSKS 指令操作数		中断时间间隔	
N	S	CX-P 设定的中断时间单位	中断时间间隔
间隔定时中断编号	中断时间		
间隔定时器 0 (中断任务 02)	#0000～#270F (0～9999)	10 ms	10～99990 ms
		1 ms	1～9999 ms
		0.1 ms	0.5～999.9 ms

外部中断功能是 CP1H PLC 的扩展功能,是指连接 CJ 系列的高功能 I/O 单元或 CJ 系列的 CPU 高功能单元时,通过单元侧的控制,指定中断任务 0~255,并执行处理。此处不做详细介绍。

例 5-3-19 图5.46是间隔定时器中断的例子,按照 30.5 ms 的间隔执行中断任务 02。

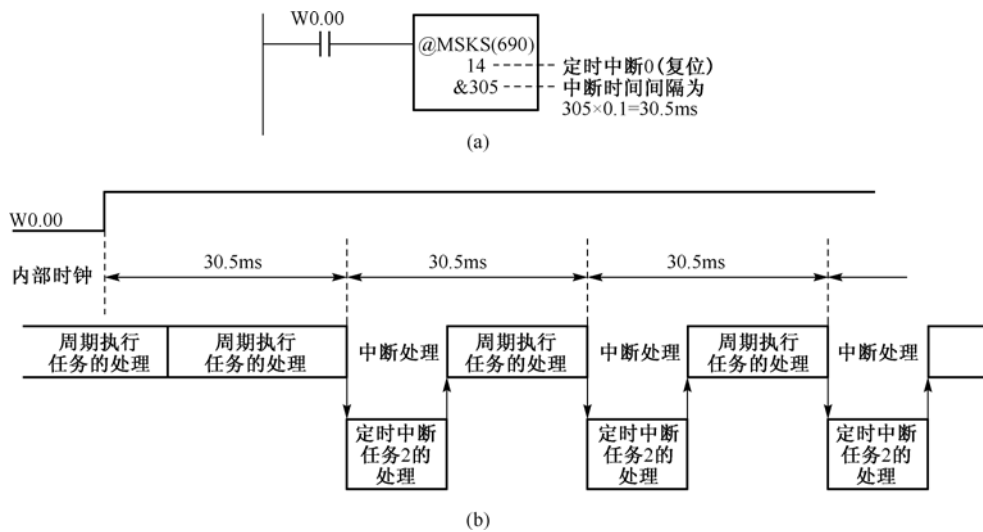


图 5.46 间隔定时器中断示例

- 设定:
- ① 通过 CX-P 的 PLC 系统设定,将定时中断单位时间设定为 0.1 ms。
  - ② 通过 CX-P 添加中断任务 02(间隔定时器 0)的新程序。
  - ③ 通过 CX-P 编制程序,如图5.47(a)所示。

图5.46(a)中@MSKS 指令的第 1 个操作数为 14,指的是使用复位开始的定时器中断功能,对应中断任务 02(间隔定时器 0);第 2 个操作数是&305,表示中断时间的间隔为 305×0.1=30.5 ms,即每隔 30.5 ms 执行一次定时中断功能。

执行条件 W0.00 为 ON 时,可进行定时中断,通过指定复位开始,将定时器复位后开始计时。每隔 30.5 ms 执行一次定时中断任务 2,动作过程如图5.46(b)所示。

5.3.6 综合例子

例 5-3-20 图5.47是中断读取 MSKR 指令的例子 1,以及执行指令后 D100 的内容。图中 MSKR 指令的第 1 个操作数为 100,指的是读取中断输入 0(0.00)的中断任务的状态;执行指令后第 2 个操作数 D100 的内容是 0003,表示中断输入 0 为加计数模式。

执行条件 W0.00 为 ON 时,读取中断输入 0 的状态,并保存在 D100 中。

例 5-3-21 图5.48是中断读取 MSKR 指令的例子 2,以及指令执行后 D200 的内容。图5.48中 MSKR 指令的第 1 个操作数为 4,表示读取间隔定时器 0(中断任务 02)的设定值;执行指令后第 2 个操作数 D200 的内容是 0069,表示定时器 0(中断任务 02)的设定值为 (0069) Hex=105。

执行条件 W0.00 为 ON 时,读取定时中断 0 的设定值,并保存在 D200 中。

例 5-3-22 图5.49是中断解除 CLI 指令的例子。

图5.49(a)是输入中断解除 CLI 指令的例子。图中 CLI 指令的第 1 个操作数为 100,指输入中断 0(0.00)的中断任务的状态;指令的第 2 个操作数是#0001,表示清除所有输入中断 0 的设置信息。图 5.49(a)中 W0.00 为 ON 时,将清除所有输入中断 0 的设置信息。

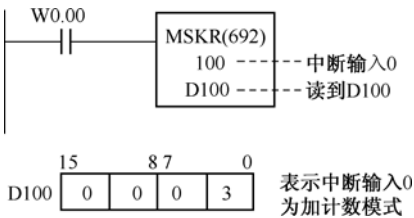


图 5.47 中断读取 MSKR 指令示例 1

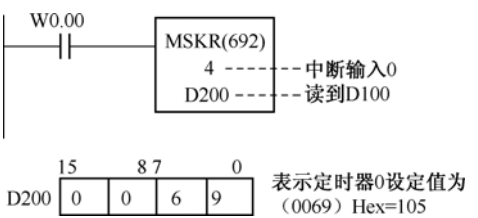


图 5.48 中断读取 MSKR 指令示例 2

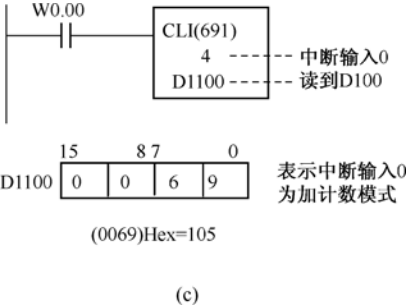
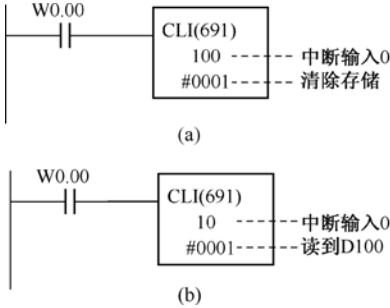


图 5.49 中断解除 CLI 指令示例

图5.49(b)是高速计数器中断解除 CLI 指令的例子。图中 CLI 指令的第 1 个操作数为 10，指高速计数器中断 0 的中断任务；指令的第 2 个操作数是#0001，表示清除所有高速计数器中断 0 的设置信息。

图5.49(c)是间隔定时器中断设定 CLI 指令的例子。图中 CLI 指令的第 1 个操作数为 4，指间隔定时器 0(0.00)；指令的第 2 个操作数 D1100 的内容是#0069，(0069)Hex=105，表示间隔定时器 0 的初次中断开始时间设定为 10.5 ms (PLC 系统设定的单位时间为 0.1 ms)。

例 5-3-23 图 5.50 是从中断输入 0.00 开始经过一定时间(0.5 ms)后的脉冲输出例子。

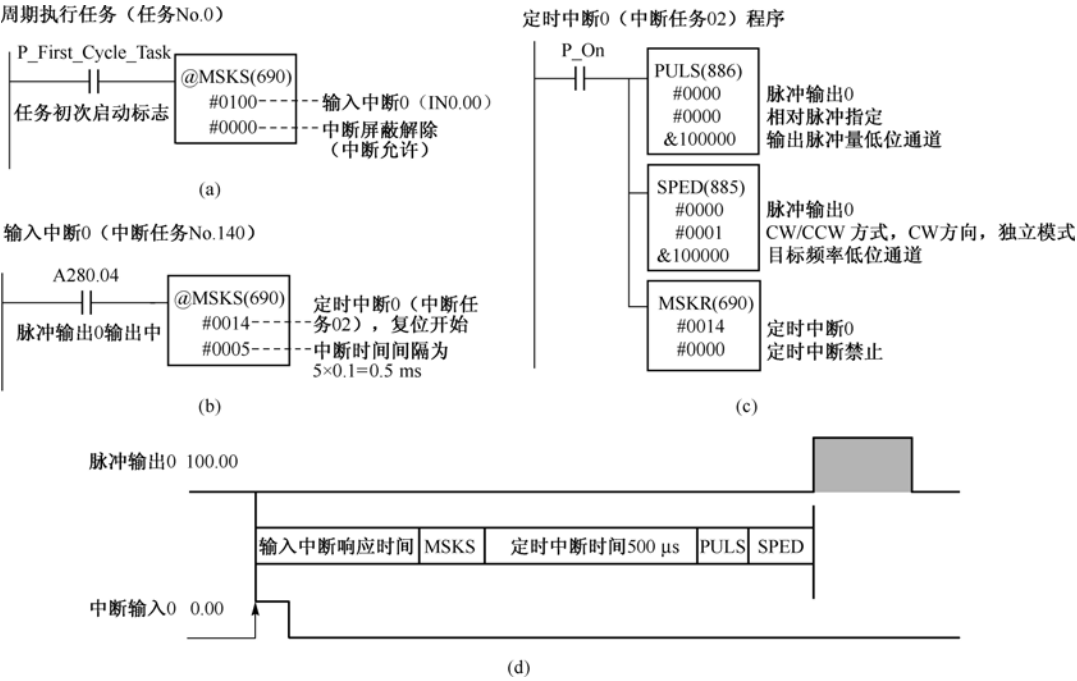


图 5.50 定时中断脉冲输出示例

设定：

- ① 在 CX-P 编程软件的工程目录下，双击项目树的“设置”，弹出“PLC 设定”对话框，在“内置输入设置”选项卡，最下方的中断输入“IN0”选为“中断”，即将输入点 0.00 设置为中断输入点。
- ② 通过 CX-P 添加中断任务 140(输入中断 0)的新程序，编制中断处理用的程序。
- ③ 通过 CX-P 添加中断任务 02(间隔定时器 0)的新程序。
- ④ 通过 CX-P 的“PLC 设定”，将定时中断单位时间设定为 0.1 ms。
- ⑤ 通过 CX-P 编制程序，如图5.50所示。

图5.50(a)输入中断解除@MSKS 指令的第 1 个操作数为#0100，指的是使用输入中断 0(IN0.00)功能，对应中断任务 140；第 2 个操作数是#0000，表示中断允许。图 5.50(a)中 P\_First\_Cycle\_Task 为 ON 时，将中断周期执行任务 0，转去执行中断任务 140(输入中断 0)。

图5.50(b)中@MSKS 指令的第 1 个操作数为 0014，指的是使用复位开始的定时器中断功能，对应中断任务 02(间隔定时器 0)；第 2 个操作数是#0005，表示中断时间的间隔为  $5 \times 0.1 = 0.5\text{ ms}$ ，即每隔 0.5 ms 执行一次定时中断功能。

图5.50(c)中用了 3 条指令。PULS 和 SPED 完成 100000 个 100 kHz 频率的脉冲的输出。MSKS 指令的第 2 操作数为#0000，表示禁止定时中断。

从中断输入 0.00 为 ON 开始经过一定时间(0.5 ms)后，从脉冲输出 0 输出 100 kHz 频率的脉冲 100000 个。作为程序，在输入中断 0 任务(中断任务 No.140)内，启动定时中断时间 0.5 ms 的定时中断。接着，在定时中断任务内执行脉冲输出的指令，同时停止定时中断，动作过程如图 5.50(d)所示。

例 5-3-24 JOG 运行是指脉冲输出的方向 CW/CCW 交替变换。

要求：如图5.51所示。

- ① 输入 1.04 为 ON 期间，在脉冲输出 1 中进行低速的 JOG 动作(CW 方向)。
- ② 输入 1.05 为 ON 期间，在脉冲输出 1 中进行低速的 JOG 动作(CCW 方向)。
- ③ 输入 1.06 为 ON 期间，在脉冲输出 1 中进行高速的 JOG 动作(CW 方向)。
- ④ 输入 1.07 为 ON 期间，在脉冲输出 1 中执行高速的 JOG 动作(CCW 方向)。

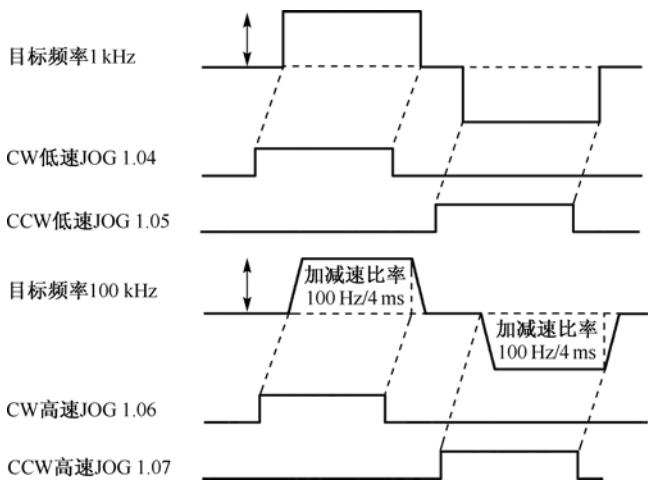


图 5.51 JOG 运行示例图

系统设定见表 5.24，程序如图5.52所示。



表 5.24 DM 区域设定(JOG 运行速度控制的设定 D0~D1、D10~D15)

设定内容	地 址	数 据
目标频率(低速)1000 Hz	D0	#03E8
	D1	#0000
加减速比率 100 Hz/4 ms	D10	#0064
目标频率(高速)100000 Hz	D11	#86A0
	D12	#0001
加减速比率 100 Hz/4 ms(不使用)	D13	#0064
目标频率(停止)0 Hz	D14	#0000
	D15	#0000

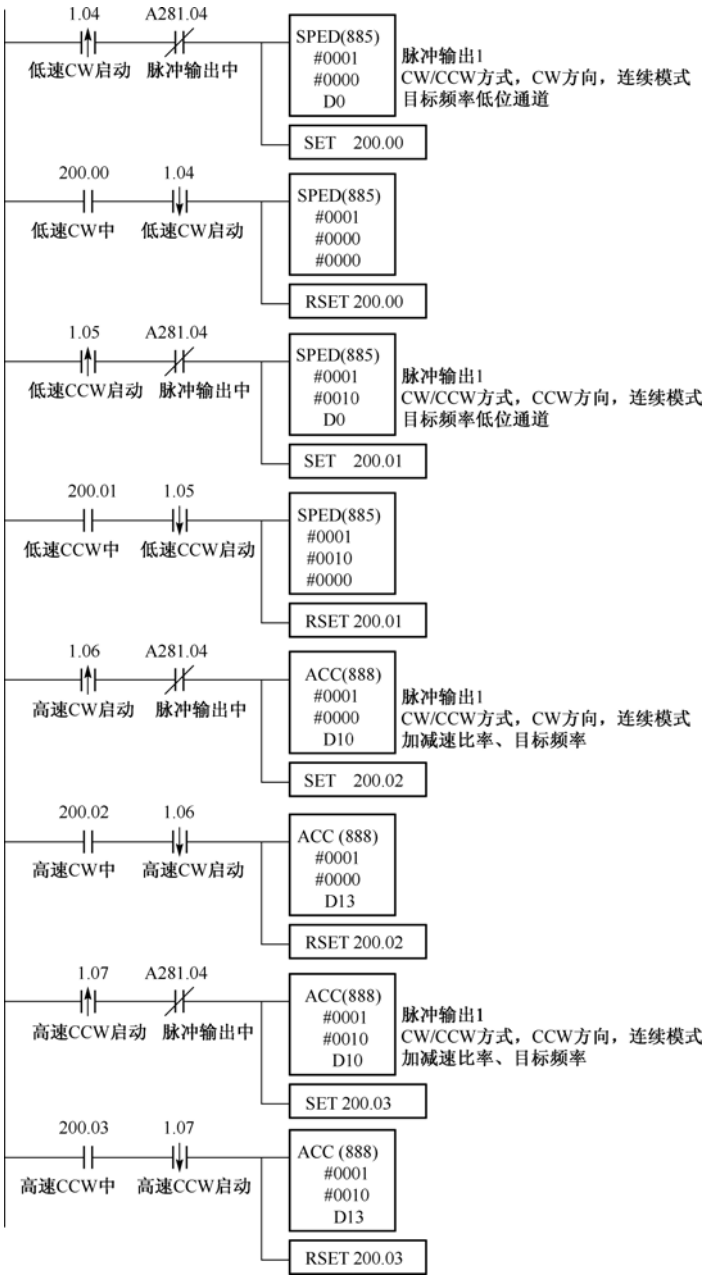


图 5.52 JOG 运行程序

## 习 题

1. CP1H PLC 任务的种类有哪些?
2. 周期执行任务与中断任务的区别是什么?
3. 周期执行任务的状态有哪些? 如何转换?
4. CP1H PLC 具有哪些中断功能?
5. 如何更改定时器中断与高速计数器中断的设定值?
6. CP1H 高速计数的输入信号有几种模式? 频率各是多少? 高速计数器使用前怎样进行设定?
7. 用高速计数器控制指令分别编写一个能实现下列控制要求的程序, 并进行必要的设定。画出梯形图, 写出语句表。
  - (1) 当高速计数器的当前值为 1500 时, 启动 A 电动机运行 5 min 后自停; 当高速计数器的当前值为 15000 时, 启动电动机 B 并运行 10 min 后自停。
  - (2) 当高速计数器的当前值大于 25500、小于 25550 时, 停止升降机的运行。
8. CP1H 有哪些中断功能? 中断优先级怎样规定? 输入中断和间隔定时器中断各有几种模式? 使用中断控制之前应怎样进行设定?
9. 用中断控制指令分别编写一个能实现下列各要求的控制程序, 并进行必要的设定。画出梯形图, 写出语句表。
  - (1) 只要 0.03 输入了一个信号, 则立即启动 CNT0001 开始计数。
  - (2) 当 0.04 输入了 100 个信号时, 则立即启动 CNT0001 开始计数。
  - (3) 当电磁阀的线圈接通 200 ms 时, 启动电动机运行。
  - (4) 每 500 ms 中, 电磁阀线圈接通 400 ms, 断 100 ms。
10. 启动按钮 0.00 置于 ON 时, 实现从脉冲输出 1 输出 50000 个脉冲, 使电动机运行的控制。要求加减速比率相同为 200 Hz/4 ms、目标频率为 500 Hz、脉冲输出量设定值为 50000 个。试编写梯形图程序。
11. 启动按钮 0.00 置于 ON 时, 实现从脉冲输出 1 输出 100000 个脉冲, 使电动机运行的控制。要求加、减速比率分别为 200 Hz/4 ms, 250 Hz/4 ms, 启动频率为 100 Hz、目标频率为 500 Hz, 脉冲输出量设定值为 100000 个。试编写梯形图程序。
12. 利用 PLC 设计一个步进电动机转速检测显示控制程序。步进电动机上装有一转速检测装置(每转输出 10 个脉冲)。电动机转速由 SPED、PULS 及 INI 等脉冲输出指令控制。控制要求如下:
  - (1) 检测电动机转速, 并存在 D 区。
  - (2) 当检测值小于给定值时, 增加脉冲频率, 使实际转速与给定值相等。

## 第6章 PLC 控制系统的设计

PLC 控制系统的设计包括三个重要的环节，其一是对控制任务的分析，确定控制系统的总体设计方案；其二是根据控制要求确定硬件构成方案；其三是设计出满足控制要求的应用程序。要想顺利地设计 PLC 控制系统，需要不断地学习和实践。本章将介绍控制系统设计的基本步骤和程序设计的基本方法。

### 6.1 概述

#### 6.1.1 PLC 控制系统设计的基本步骤

PLC 控制系统的设计一般包括下面几个基本步骤。

##### 1. 对控制任务做深入的调查研究

在着手设计之前，要详细了解工艺过程和控制要求，例如弄清哪些信号须输入 PLC，是模拟量还是开关量，应采取什么方式，选用什么元件输入信号；哪些信号须从 PLC 输出到外部，通过什么执行元件去驱动负载；弄清整个工艺过程各个环节相互的联系；了解机械运动部件的驱动方式，是液压、气动还是电动，以及运动部件与各电气执行元件之间的联系；了解系统的控制是全自动还是半自动的，控制过程是周期性，还是单周期运行，是否有手动调整要求等。另外要注意哪些量需要监控、报警、显示，是否需要故障诊断，需要哪些保护措施等。

##### 2. 确定系统总体设计方案

这是最为重要的一步。若总体方案的决策有误，则整个设计任务不能顺利完成，甚至失败并造成很大的投资浪费。要在全面、深入了解控制要求的基础上确定电气控制方案。

##### 3. 根据控制要求确定输入、输出元件，选择 PLC 机型

在确定电气控制方案之后，可进一步研究系统的硬件构成。要选择合适的输入和输出元件，确定主回路各电器及保护器件，选择报警、显示元件等。根据所选用的电器、元件的类型和数量，计算所需 PLC 的输入/输出点数，并参照其他要求选择合适的 PLC 机型。

##### 4. 确定 PLC 的输入/输出点分配

明确各输入电气与 PLC 输入点的对应关系，以及 PLC 各输出点与各输出执行元件的对应关系，写出 PLC 的 I/O 分配表。

##### 5. 设计应用程序

在完成上述工作之后可开始控制系统的程序设计。程序设计的质量关系到系统运行的稳定性和可靠性。应根据控制要求拟订几个设计方案，经认真比较后选择出最佳编程方案。当控制系统较复杂时，可将其分成多个相对独立的任务，最后将各任务的程序合理地连接在一起。

##### 6. 应用程序的调试

对编好的程序，可以先利用模拟实验板模拟现场信号进行初步调试。经反复调试、修改后，使程序基本满足控制要求。

## 7. 制作电气控制柜和控制盘

在系统硬件构成方案确定之后,就可以考虑电气控制柜及控制盘(或称操作盘)的设计和制作。在动手制作之前,要画出电气控制主回路电路图。在设计主回路时要全面地考虑各种保护、连锁等问题。在布置控制柜和布线时,要采取有效的措施抑制各种干扰信号,同时注意防尘、防静电、防雷电等问题。

## 8. 联机调试程序

联机调试时才能发现程序存在的实际问题和不足,通过调试、修改后,使程序完全符合控制要求。调试前要制定周密的调试计划,以免由于工作的盲目性而隐藏了应该发现的问题。另外,程序调试完毕必须经过一段时间运行的考验,才能确认程序是否达到控制要求。

## 9. 编写技术文件

这部分工作包括整理程序清单并保存程序,编写元件明细表,绘制电气原理图及主回路电路图,整理相关的技术参数,编写控制系统说明书等。

# 6.1.2 PLC 的应用程序

## 1. 应用程序的内容

应用程序应最大限度地满足系统控制功能的要求,在构思程序主体的框架后,要以它为主线,逐一编写实现各控制功能或各任务的程序,经过不断调整、完善,使程序能完成指定的功能。通常应用程序还应包括以下几个内容。

### (1) 初始化程序

在 PLC 加电后,一般都要做一些初始化的操作。其作用是为启动做必要的准备,并避免系统发生误动作。初始化程序的主要内容:将某些数据区、计数器清零,使某些数据区恢复所需数据,对某些输出位置位或复位,显示某些初始状态等。

### (2) 检测、故障诊断、显示程序

应用程序一般都设有检测、故障诊断和显示等内容。这些内容可以在程序设计基本完成时再进行添加。有时,它们也是相对独立的程序段。

### (3) 保护、连锁程序

各种应用程序中,保护和连锁是不可缺少的部分。它可以杜绝由于非法操作而引起的逻辑控制混乱,保证系统的运行更安全、可靠,因此要认真考虑保护和连锁的问题。通常在 PLC 外部也要设置连锁和保护措施。

## 2. 应用程序的质量

对同一个控制要求,即使选用同一个机型的 PLC,用不同设计方法所编写的程序,其结构也可能不同。尽管几种程序都可以实现同一控制功能,但是程序的质量却可能差别很大。程序的质量可以由以下几个方面来衡量。

### (1) 程序的正确性

应用程序的好坏,最根本的一条就是正确。所谓正确的程序必须能经得起系统运行的考验,离开这一条对程序所做的评价都是没有意义的。

### (2) 程序的可靠性好

好的应用程序,可以保证系统在正常和非正常(短时掉电再复电、某些被控量超标、某个环节有故障等)工作条件下都能安全可靠地运行,也可保证在出现非法操作(如按动或误触动了不该动作的按钮)等情况下不至于出现系统控制失误。

(3) 参数的易调整性好

PLC 控制的优越性之一就是灵活性好，容易通过修改程序或参数而改变系统的某些功能。例如，有的系统在一定情况下需要变动某些控制量的参数(如定时器或计数器的设定值等)，在设计程序时必须考虑怎样编写才能易于修改。

(4) 程序要简练

编写的程序应尽可能简练，减少程序的语句，一般可以减少程序扫描时间，提高 PLC 对输入信号的响应速度。当然，如果过多地使用那些执行时间较长的指令，有时虽然程序的语句较少，但是其执行时间也不一定短。

(5) 程序的可读性好

程序不仅仅给编者自己看，系统的维护人员也要读。另外，为了有利于交流，也要求程序有一定的可读性。

要想顺利地完成控制系统的设计，不仅要熟练掌握各种指令的功能及使用规则，还要学习如何编程，下面介绍如下常用的几种编程方法：逻辑设计法、时序图设计法、顺序控制设计法、经验设计法等。

6.2 程序设计方法

6.2.1 逻辑设计法

当主要对开关量进行控制时，使用逻辑设计法比较好。逻辑设计法的基础是逻辑代数。在程序设计时，对控制任务进行逻辑分析和综合，将控制电路中元件的通、断电状态视为以触点通、断状态为逻辑变量的逻辑函数，对经过化简的逻辑函数利用 PLC 的逻辑指令可以顺利地设计出满足要求的、较为简练的控制程序。这种方法设计思路清晰，所编写的程序易于优化，是一种较为实用、可靠的程序设计方法。下面以一个简单的控制为例介绍这种编程方法。

**例 6-2-1** 某系统中有 4 台通风机，要求在以下几种运行状态下发出不同的显示信号：3 台及 3 台以上开机时，绿灯常亮；两台开机时，绿灯以 5Hz 的频率闪烁；1 台开机时，红灯以 5Hz 的频率闪烁；全部停机时，红灯常亮。

由控制任务可知，这是一个对通风机运行状态进行监视的问题。显然，必须把 4 台通风机的各种运行状态的信号输入到 PLC 中(由 PLC 外部的输入电路来实现)，各种运行状态对应的显示信号是 PLC 的输出。

为了讨论问题方便，设 4 台通风机分别为 A、B、C、D，红灯为 F1，绿灯为 F2。由于各种运行情况所对应的显示状态是唯一的，故可将几种运行情况分开进行程序设计。

1. 红灯常亮的程序设计

当 4 台通风机都不开机时红灯常亮。设灯常亮为“1”、灭为“0”，通风机开机为“1”、停为“0”(下同)。其状态表为：

A	B	C	D	F1
0	0	0	0	1

由状态表可得 F1 的逻辑函数：

$$F1 = \overline{A}\overline{B}\overline{C}\overline{D}$$

根据逻辑函数(1)画出其梯形图，如图6.1所示。

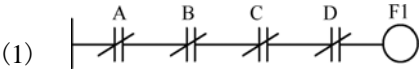


图 6.1 红灯常亮的梯形图

2. 绿灯常亮的程序设计

能引起绿灯常亮的情况有 5 种，列状态表为：

A	B	C	D	F2
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

由状态表可得 F2 的逻辑函数为：

$$F2 = \bar{A}BCD + A\bar{B}CD + AB\bar{C}D + ABC\bar{D} + ABCD$$

(2)

根据这个逻辑函数直接画梯形图时，梯形图会很繁琐，所以要先对逻辑函数 (2) 进行化简。例如将式 (2) 化简成下式：

$$F2 = AB(D + C) + CD(A + B)$$

(3)

再根据式 (3) 画出的梯形图如图 6.2 所示。

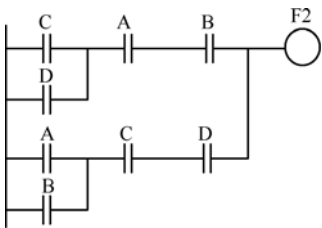


图 6.2 绿灯常亮的梯形图

3. 绿灯闪烁的程序设计

绿灯闪烁是绿灯亮的状态加一个脉冲。所以绿灯闪烁为“1”，列状态表为

A	B	C	D	F2
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1

由状态表可得 F2 的逻辑函数为

$$F2 = \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}D + AB\bar{C}\bar{D}$$

(4)

将式 (6) 化简为

$$F2 = (\bar{A}B + A\bar{B})(\bar{C}D + C\bar{D}) + \bar{A}BCD + AB\bar{C}\bar{D}$$

(5)

根据式 (7) 画出其梯形图如图 6.3 所示。

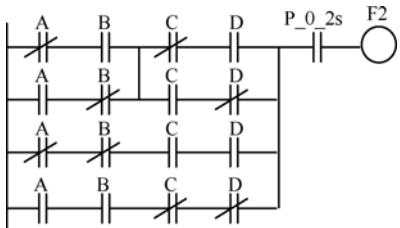


图 6.3 绿灯闪烁的梯形图

4. 红灯闪烁的程序设计

红灯闪烁实际是红灯亮的状态加一个脉冲。所以红灯闪烁为“1”，列状态表为：

A	B	C	D	F1
0	0	0	1	1
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1

由状态表可得 F1 的逻辑函数为

$$F1 = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D}$$

(6)

将式 (4) 化简为

$$F1 = \bar{A}\bar{B}(\bar{C}D + C\bar{D}) + \bar{C}\bar{D}(\bar{A}B + A\bar{B})$$

(7)

由式 (5) 画出的梯形图如图 6.4 所示。其中 P\_0.2s 能产生 0.2s 即 5 Hz 的脉冲信号。

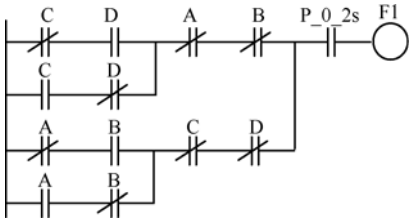


图 6.4 红灯闪烁的梯形图

5. 选择 PLC 机型、做 I/O 点分配

本例只有 A、B、C、D 四个输入信号，F1、F2 两个输出，若系统选择的机型是 CP1H，做出 I/O

分配为：

输 入				输 出	
A	B	C	D	F1	F2
1.01	1.02	1.03	1.04	101.01	101.02

由 I/O 分配及图 6.1～图 6.4，综合在一起便得到总梯形图如图 6.5 所示。

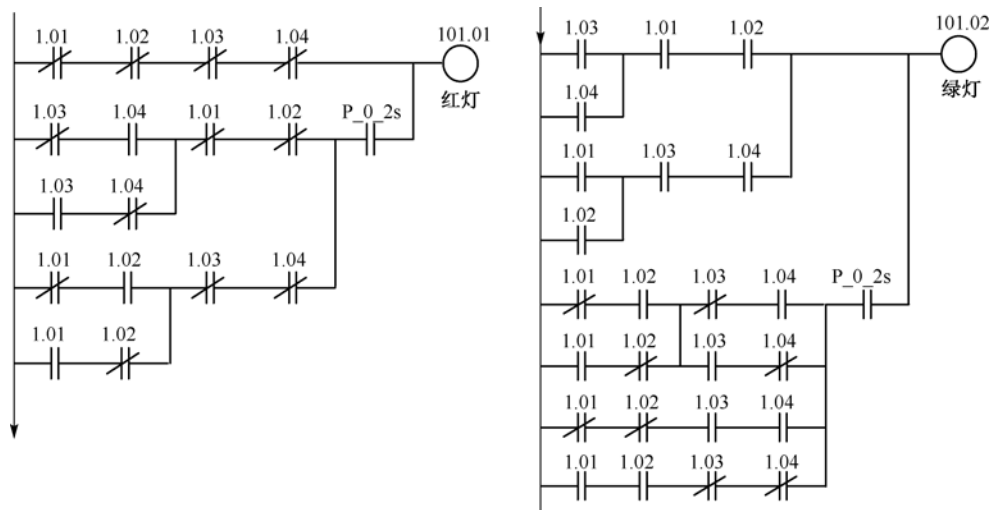


图 6.5 通风机运行状态显示的梯形图

下面把逻辑设计法归纳如下。

- ① 用不同的逻辑变量来表示各输入、输出信号，并设定对应输入、输出信号各种状态时的逻辑值。
- ② 根据控制要求，列出状态表或画出时序图。
- ③ 由状态表或时序图写出相应的逻辑函数，并进行化简。
- ④ 根据化简后的逻辑函数画出梯形图。
- ⑤ 上机调试，使程序满足要求。

6.2.2 时序图设计法

如果 PLC 各输出信号的状态变化有一定的时间顺序，可用时序图法设计程序。因为在画出各输出信号的时序图后，容易理顺各状态转换的时刻和转换的条件，从而建立清晰的设计思路。下面通过一个例子说明这种设计方法。

**例 6-2-2** 在十字路口上设置的红、黄、绿交通信号灯，其布置如图6.6所示。由于东西方向车流量较小，南北方向车流量较大，所以南北方向的放行(绿灯亮)时间为 30 s，东西方向的放行时间(绿灯亮)为 20 s。当在东西(或南北)方向的绿灯灭时，该方向的黄灯与南北(或东西)方向的红灯一起以 5 Hz 的频率闪烁 5 s，以提醒司机和行人注意。闪烁 5 s 之后，立即开始另一个方向的放行。要求只用一个控制开关对系统进行启停控制。

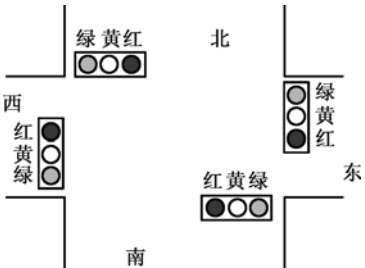


图 6.6 交通灯布置图

下面介绍用时序图法编程的思路。

- ① 分析 PLC 的输入和输出信号,以作为选择 PLC 机型的依据

之一。在满足控制要求的前提下，应尽量减少占用 PLC 的 I/O 点。由上述控制要求可见，由控制开关输入的启、停信号是输入信号。由 PLC 的输出信号控制各指示灯的亮、灭。在图 6.6 中，南北方向的三色灯共 6 盏，同颜色的灯在同一时间亮、灭，所以可将同色灯两两并联，用一个输出信号控制，这样南北方向交通灯占用 3 个输出点。同理，东西方向的三色灯也照此办理，占用 3 个输出点。所有交通灯共占 6 个输出点。

② 根据控制要求，画出各方向三色灯的工作时序图，如图 6.7 所示。

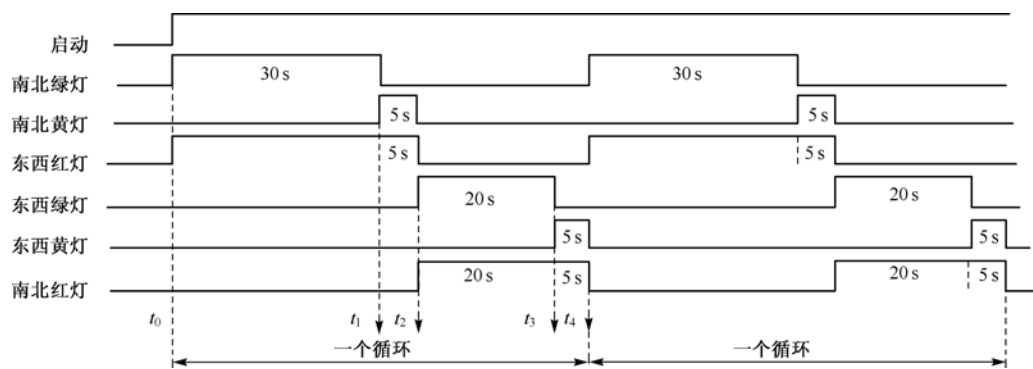


图 6.7 交通灯工作时序图

③ 由时序图分析各输出信号之间的时间关系。图 6.7 中，南北方向放行时间可分为两个时间区段，南北方向的绿灯和东西方向的红灯亮，换行前南北方向的黄灯与东西方向的红灯一起闪烁；东西方向放行时间也分为两个时间区段，东西方向的绿灯和南北方向的红灯亮，换行前东西方向的黄灯与南北方向的红灯一起闪烁。一个循环内分为 4 个区段，这 4 个时间区段对应着 4 个分界点： $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$ 。在这 4 个分界点处信号灯的状态将发生变化。

④ 4 个时间区段必须用 4 个定时器来控制，为了明确各定时器的职责，以便于理顺各色灯状态转换的准确时间，最好列出定时器的功能明细表，见表 6.1。

表 6.1 各定时器一个循环中的功能明细表

定时器	$t_0$ (0 s)	$t_1$ (30 s)	$t_2$ (35 s)	$t_3$ (55 s)	$t_4$ (60 s)
TIM0000 定时 30 s	开始定时 南北绿、东西红灯亮	定时到，输出 ON 且保持 南北绿灯灭，南北黄、东西红灯闪	ON	ON	开始下一个循环的定时
TIM0001 定时 35 s	开始定时	继续定时	定时到，输出 ON 且保持 南北黄、东西红灯灭，东西绿、南北红灯亮	ON	开始下一个循环的定时
定时器	$t_0$ (0 s)	$t_1$ (30 s)	$t_2$ (35 s)	$t_3$ (55 s)	$t_4$ (60 s)
TIM0002 定时 55 s	开始定时	继续定时	继续定时	定时到，输出 ON 且保持 东西绿灯灭，东西黄、南北红灯闪	开始下一个循环的定时
TIM0003 定时 60 s	开始定时	继续定时	继续定时	继续定时	定时到，输出 ON，随即自复位且开始下一个循环的定时 东西黄、南北红灯灭，



					南北绿、东西红灯亮
--	--	--	--	--	-----------

⑤ 进行 PLC 的 I/O 分配。下面是使用 CP1H 时所做的 I/O 分配，见表 6.2。

表 6.2 I/O 分配

输 入	输 出					
控制开关	南北绿灯	南北黄灯	南北红灯	东西绿灯	东西黄灯	东西红灯
0.00	100.00	100.01	100.02	100.03	100.04	100.05

⑥ 根据定时器功能明细表和 I/O 分配，画出的梯形图如图 6.8 所示。对图 6.8 的设计意图及功能简要分析如下。

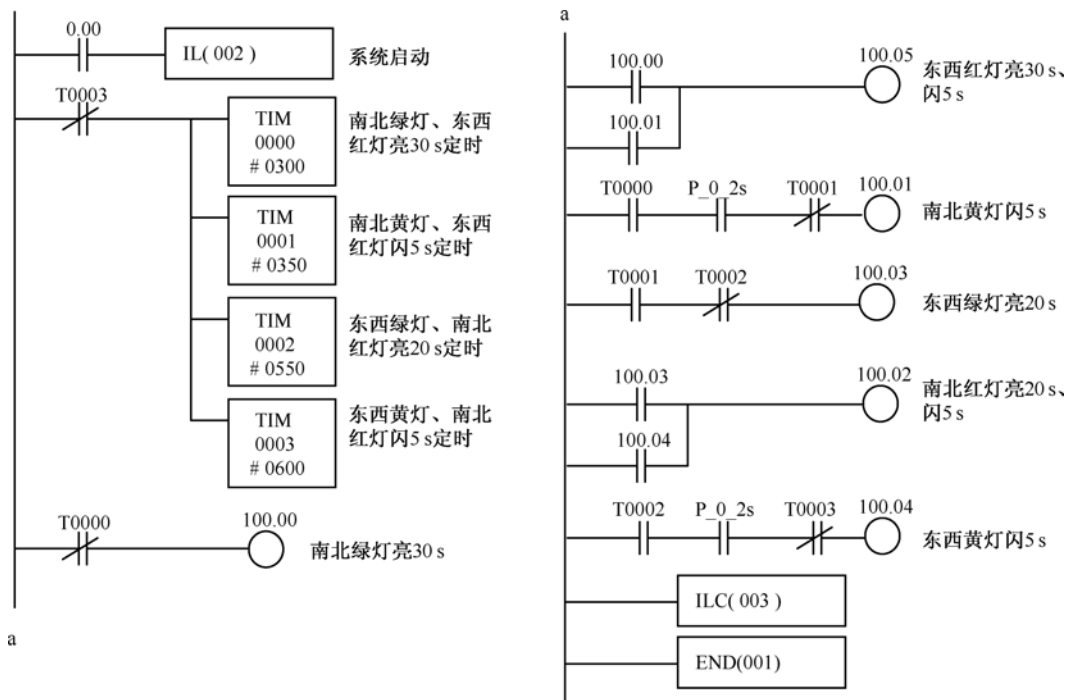


图 6.8 交通信号灯控制梯形图

- 程序用 IL/ILC 指令控制系统启、停，当 0.00 为 ON 时程序执行，否则不执行。
- 程序启动后 4 个定时器同时开始定时，100.00 为 ON，使南北绿灯亮、东西红灯亮。
- 当 TIM0000 定时时间到，其一，100.00 为 OFF 使南北绿灯灭；其二，100.01 为 ON 使南北黄灯闪烁 (P\_0\_2s 以 5 Hz 的频率 ON、OFF)、东西红灯也闪烁。
- 当 TIM0001 定时时间到，其一，100.01 为 OFF 使南北黄灯、东西红灯灭；其二，100.03 为 ON 使东西绿灯、南北红灯亮。
- 当 TIM0002 定时时间到，其一，100.03 为 OFF 使东西绿灯灭；其二，100.04 为 ON 使东西黄灯闪烁、南北红灯也闪烁。
- TIM0003 记录一个循环的时间。当 TIM0003 定时时间到，其一，100.04 为 OFF 使东西黄灯、南北红灯灭；其二，TIM0000~TIM0003 全部复位，并开始下一个循环的定时。由于 TIM0000 为 OFF，所以南北绿灯亮、东西红灯亮，并重复上述过程。

下面把时序图设计法归纳如下：

- ① 详细分析控制要求，明确各输入、输出信号个数，合理选择机型。
  - ② 明确各输入和各输出信号之间的时序关系，画出各输入和输出信号的工作时序图。
  - ③ 把时序图划分成若干个时间区段，确定各区段的时间长短。找出区段间的分界点，弄清分界点处各输出信号状态的转换关系和转换条件。
  - ④ 根据时间区段的个数确定需要几个定时器，分配定时器号，确定各定时器的设定值，明确各定时器开始定时和定时时间到这两个关键时刻对各输出信号状态的影响。
  - ⑤ 对 PLC 进行 I/O 分配。
  - ⑥ 根据定时器的功能明细表、时序图和 I/O 分配画出梯形图。
  - ⑦ 做模拟运行实验，检查程序是否符合控制要求，进一步修改程序。
- 对于一个复杂的控制系统，若某个环节属于这类控制，就可以用这个方法去处理。

6.2.3 顺序控制设计法

对那些按动作的先后顺序进行控制的系统，非常适宜使用顺序控制设计法编程。顺序控制设计法规律性很强，虽然编出的程序偏长，但程序结构清晰、可读性好。

1. 功能图

在用顺序控制设计法编程时，功能图是很重要的工具。功能图能清楚地表现出系统各工作步的功能、步与步之间的转换顺序及转换条件。

(1) 功能图的组成

以下面简单的控制为例来说明功能图的组成。

某动力头的运动状态有三种，即快进→工进→快退。各状态的转换条件为：快进到一定位置压限位开关 ST1 则转为工进，工进到一定位置压限位开关 ST2 则转为快退，退回原位压 ST3，动力头自动停止运行。对这样的控制过程画出的功能图如图6.9所示。

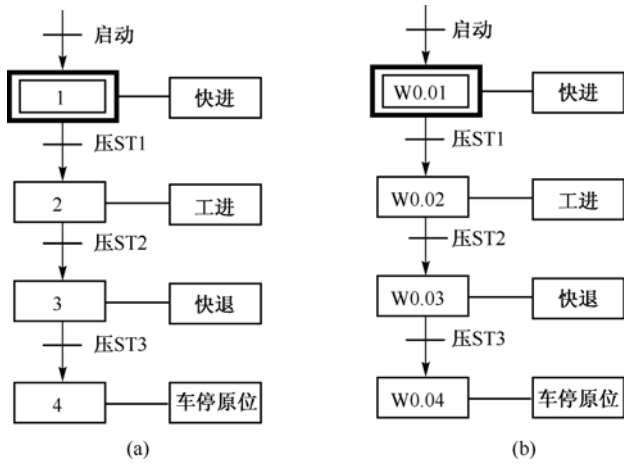


图 6.9 动力头顺序功能图

功能图是由步、有向连线、转换与转换条件和动作等组成的。

① 步。

功能图设计法是将系统的一个工作周期划分为若干个顺序相连的阶段，这些阶段称为步。步是顺序功能图的最基本组成部分，它是某一特定控制功能的程序段。与系统的初始状态相对应的步称为“初始步”，初始状态一般是系统等待启动命令的相对静止的状态。用矩形框表示各步，框内的数字是步的

编号。初始步使用双线框,图6.9(a)是步的顺序功能图,其中步1就是初始步,每个功能图都有一个初始步。图6.9(b)是控制位(对应步)的顺序功能图。控制位W0.01~W0.04分别代表步1~步4。初始步没有动作为等待步时,以0步开始命名各步;当初始步有动作时,从1步开始命名各步。

### ② 动作。

步是某一特定控制功能的程序段,在每一步中,要执行相应的控制功能和动作,在每一步的右边用矩形框中的内容来表示与该步相对应的动作,该矩形框应与对应步的矩形框相连。每一步可能是一个动作,如图6.9所示,步1的动作是快进。也可能包含几个动作。若某一步包含几个动作,可以选用图6.10中的画法来表示,但是并不表示这些动作之间存在着任务顺序。说明动作的语句应清楚地表明

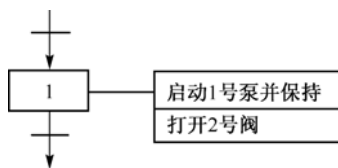


图 6.10 单步多动作画法示例

该动作是存储型的还是非存储型的。例如某步的存储型动作“启动1号泵并保持”,是指该步活动时开启1号泵,该步不活动时继续开着;而非存储型动作“打开2号阀”,是指该步活动时打开,不活动时关闭。

### ③ 有向连线。

步与步之间用有向线段相连,箭头表示步的转换方向(简单的功能图可不画箭头)。

### ④ 转换与转换条件。

转换是步与步之间活动状态的传递,转换用步与步之间的与有向线段垂直的短横线表示。短横线旁标注转换条件。转换条件是使系统由当前步进入下一步的信号,正在执行的步称为活动步,当前一步为活动步且转换条件满足时,将启动下一步并终止前一步的执行。转换条件可以是外部输入信号,如按钮、开关、限位开关的通/断等;也可以是PLC内部产生的信号,如定时器、计数器的触点提供的信号;还可以是若干信号的组合。

顺序控制(功能图)设计法是指用转换条件控制各步的编程元件,让它们的状态按一定的顺序变化,从而达到控制PLC各输出位的目的。

### (2) 功能图的类型

功能图从结构上来分,可分为单序列、选择序列、并行序列、跳转序列、循环序列等结构。

#### ① 单序列结构。

图6.9是单序列结构类型。这种结构的功能图没有分支,每个步后只有一个步,步与步之间只有一个转换条件。

#### ② 选择序列结构。

图6.11是选择序列结构的功能图。选择序列的开始称为分支,如图中的步1之后有3个分支(或更多),各选择分支不能同时执行。例如,当步1为活动步且条件a满足时则转向步2,当步1为活动步且条件b满足时则转向步3,当步1为活动步且条件c满足时则转向步4。无论步1转向哪个分支,当其后续步成为活动步时,步1自动变为不活动步。

当已选择了转向某一个分支,则不允许另外几个分支的首步成为活动步,所以应该使各选择分支之间连锁。选择序列的结束称为合并。如图6.11中,不论哪个分支的最后一步成为活动步,当转换条件满足时都要转向步5。

#### ③ 并行序列结构。

图6.12是并行序列结构的功能图。并行序列的开始也称为分支,为了区别于选择序列结构的功能图,用双线来表示并行序列分支的开始,转换条件放在双线之上。如图中的步1之后有3个并行分支,当步1为活动步且条件a满足时,则步2、3、4同时被激活变为活动步,而步1则变为不活动步。图中步2和步5、步3和步6、步4和步7是3个并行的单序列。

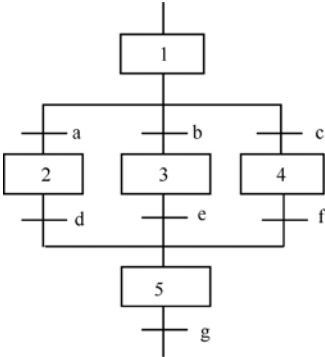


图 6.11 选择序列结构

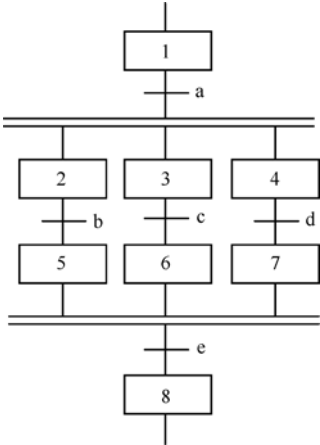


图 6.12 并行序列结构

并行序列的结束称为合并，用双线表示并行序列的合并，转换条件放在双线之下。对图 6.12，当各并行序列的最后一步即步 5、6、7 都为活动步且满足条件 e 时，将同时转换到步 8，且步 5、6、7 同时都变为不活动步。

④ 跳转序列结构。

在图 6.13 所示的结构中，当步 2 为活动步时，如果转换条件 c 满足，则按原顺序执行；如果转换条件 f 满足，则序列的进展将跳过步 3 与步 4，直接激活步 5。从原理上看，跳转序列可以作为选择序列的一种特例。

⑤ 循环序列结构。

图 6.14(a) 所示部分循环序列结构，用于描述某一段程序的多次重复执行。而图 6.14(b) 表示完整性循环结构，它用于描述整个进程的重复执行。例如，按相同尺寸加工零件，每次只加工 1 个成品，就是这种情况。

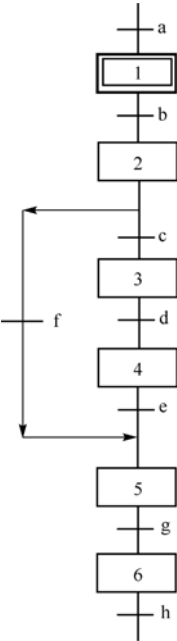


图 6.13 跳转序列结构

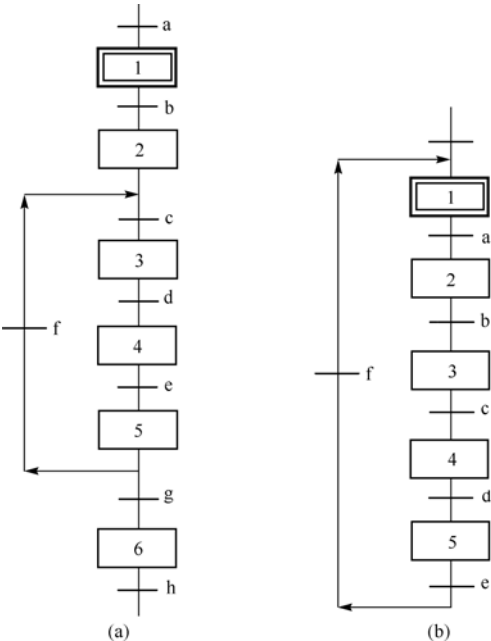


图 6.14 循环序列结构

(3) 功能图与梯形图的对应关系

由第 3 章介绍的步进控制指令的使用方法可以联想到，每个步可设置一个控制位，当某步的控制位为 ON 时，该步成为活动步（激活下一步的条件之一），同时与该步对应的程序开始执行；当转换条件满足时（激活下一步的条件之二），下一步的控制位为 ON，而上一步的控制位变为 OFF，且上一步对应的程序停止执行。显然，只要在顺序上相邻的控制位之间进行联锁，就可以实现这种步进控制。

图 6.15 是步程序的结构。线圈  $S_{i-1}$ 、 $S_i$ 、 $S_{i+1}$ 、 $S_{i+2}$  等是各步的控制位， $C_i$ 、 $C_{i+1}$ 、 $C_{i+2}$  是各步的转换条件。由上述分析可知，某一步成为活动步的条件是，前一步是活动步且转换条件满足。所以图中将常开触点  $S_{i-1}$  和  $C_i$ ，以及  $S_i$  和  $C_{i+1}$  相串联作为步启动的条件。由于转换条件是短信号，因此每步要加自锁。当后续步成为活动步时，前一步要变为不活动步，所以图中将常闭触点  $S_{i+1}$  和  $S_{i+2}$  与前一步的控制位线圈相串联。

当某一步成为活动步时，其控制位为 ON，这个 ON 信号可以控制输出继电器以实现相应的控制，例如图 6.15 中的 B1 和 B2。

(4) 根据功能图画梯形图方法

图 6.16 的功能图总体上是并行的，其中包括了一个单序列和一个选择序列。以该图为例，说明由功能图画梯形图的方法，图中矩形框内直接标以控制位，每个控制位对应一步。

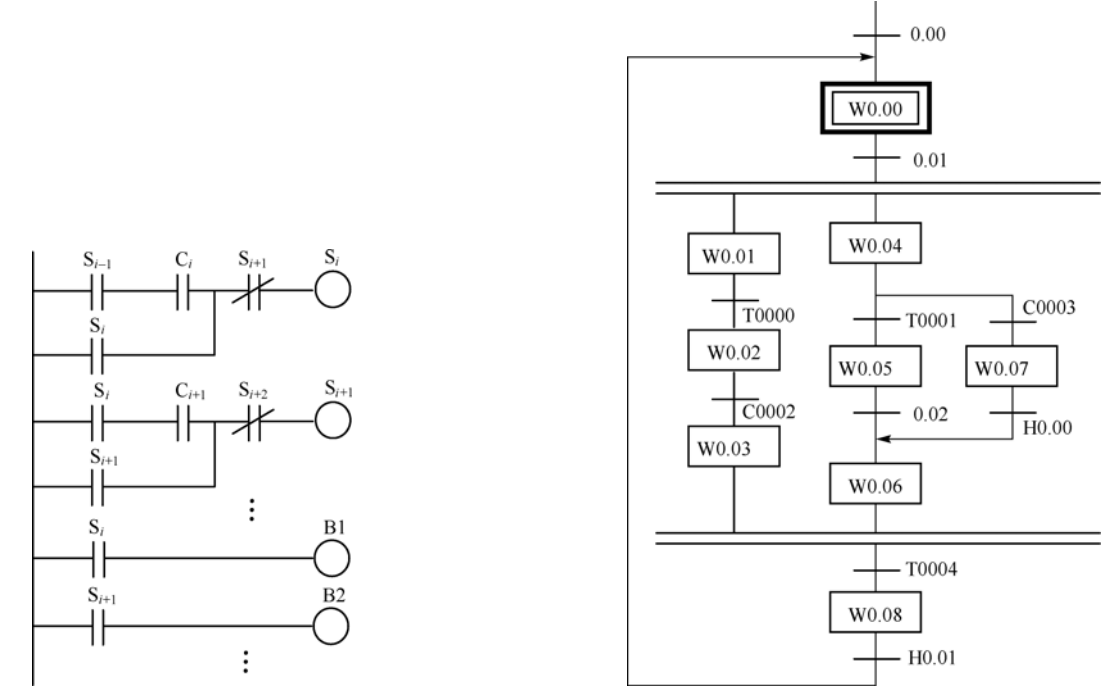


图 6.15 步程序的结构

图 6.16 功能图

① 步 W0.00。

该步为初始步，它是前面两个选择分支的合并步。因此，使 W0.00 成为活动步的条件是，或 0.00 为 ON，或步 W0.08 为活动步且 H0.01 为 ON。当步 W0.01 和 W0.04 成为活动步时，步 W0.00 变为不活动步。所以把常闭触点 W0.01 或 W0.04 与步 W0.00 的控制位线圈相串联，再加上本位的自锁，画出的梯形图如图 6.17(a) 所示。

② 步 W0.01、步 W0.02 和步 W0.03。

W0.01 是单序列的开始步，其成为活动步的条件是，步 W0.00 为活动步且转换条件 0.01 为 ON。

当步 W0.02 成为活动步时，步 W0.01 变为不活动步，所以把常闭触点 W0.02 与步 W0.01 的线圈相串联，再加上本位的自锁，画出的梯形图如图6.17(b)所示。

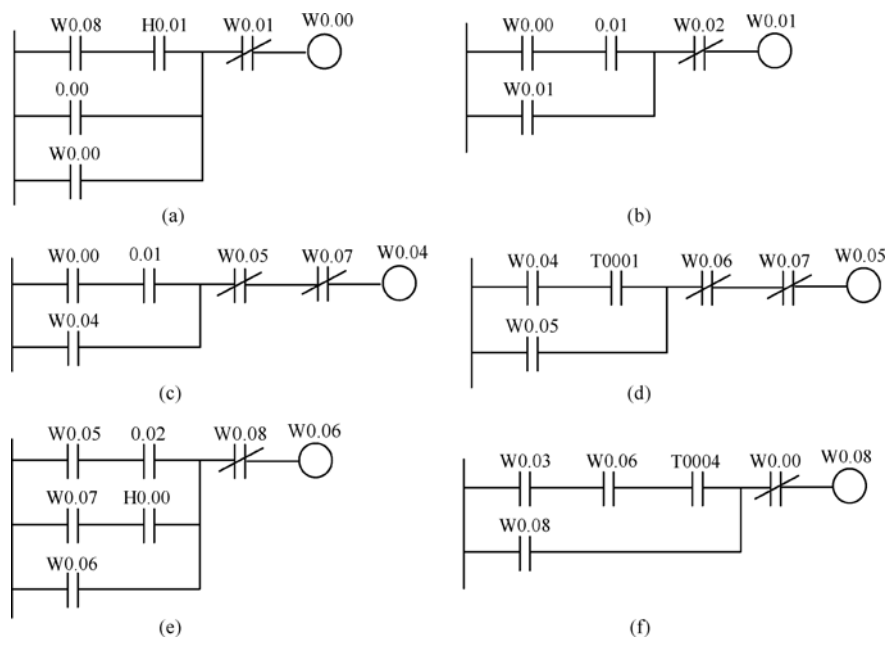


图 6.17 对应图 6.16 各步的梯形图

步 W0.02 和步 W0.03 的梯形图与步 W0.01 相似，读者自己练习画出它们的梯形图。

③ 步 W0.04。  
步 W0.04 是选择序列的开始,该步后续是两个选择分支。步 W0.04 的梯形图与步 W0.01 相似，但其线圈要与常闭触点 W0.05 和 W0.07 相串联。这是因为，无论选择了哪个分支，即不论步 W0.05 还是步 W0.07 成为活动步，步 W0.04 都要成为不活动步，画出梯形图如图 6.17(c) 所示。

④ 步 W0.05 和 W0.07。  
步 W0.05 的梯形图如图6.17(d)所示。线圈 W0.05 与常闭触点 W0.06 和 W0.07 相串联，这样，如果步 W0.07 已经成为活动步，即使步 W0.05 的条件满足也不会成为活动步，从而实现了步 W0.05 和步 W0.07 之间，即两个选择分支之间的联锁。步 W0.07 的梯形图与步 W0.05 相似，只是其转换条件是 W0.04 和 C0003 的串联，其线圈要与常闭触点 W0.06 和 W0.05 相串联。

⑤ 步 W0.06。  
步 W0.06 是选择分支的合并步。步 W0.06 成为活动步的条件是：或步 W0.05 为活动步且 0.02 为 ON，或 W0.07 为活动步且 H0.00 为 ON，所以这两个条件之间是“或”的关系。当 W0.08 为活动步时，步 W0.06 要变为不活动步，所以 W0.06 的线圈要与常闭触点 W0.08 串联。画出的梯形图如图6.17(e)所示。

⑥ 步 W0.08。  
步 W0.08 是并行序列的合并步，其成为活动步的条件为，步 W0.03 和步 W0.06 均为活动步，且 T0004 为 ON，三个条件是“与”的关系。当 W0.00 成为活动步时其变为不活动步，所以 W0.08 的线圈要与常闭触点 W0.00 串联，画出的梯形图如图6.17(f)所示。

2. 顺序控制设计法

用顺序控制设计法编程的基本步骤如下。

① 分析控制要求，弄清顺序控制的类型，将控制过程分成若干个工作步，明确每个工作步的功能，确定步的转换条件(可能是多个信号的“与”、“或”等逻辑组合)。

② 为每个步设定控制位。控制位最好使用同一个通道的若干连续位，本书采用 W0 通道的各位作为工作步(控制位)。若用定时器/计数器的输出作为转换条件，则应确定各定时器/计数器的编号和设定值。

③ 确定所需输入和输出点的个数，选择 PLC 机型，做出 I/O 分配。

④ 在前两步的基础上，画出功能图。

⑤ 根据功能图画梯形图。

⑥ 添加某些特殊要求的程序。

**例 6-2-3 送料小车的控制。**送料小车工作示意图如图 6.18 所示，初始状态小车空车停在行程开关 ST1 处，按一下启动按钮，小车在 ST1 处装料；15 s 后装料结束，开始右行；碰到行程开关 ST2 后停下来卸料；10 s 后左行，碰到行程开关 ST1 又停下来装料。这样循环工作，直到按一下停止按钮，小车在完成最后一个周期的工作后空车停在 ST1 处。

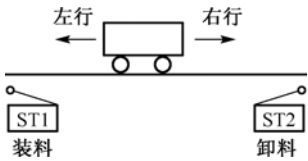


图 6.18 送料小车工作示意图

① 系统的 I/O 分配见表 6.3(带步分配)。

表 6.3 I/O 分配表

输 入		输 出		步	
ST1	0.01	右行接触器	100.00	0 步	W0.00
ST2	0.02	左行接触器	100.01	1 步	W0.01
启动按钮	0.04	装料电磁阀	100.02	2 步	W0.02
停止按钮	0.05	卸料电磁阀	100.03	3 步	W0.03
—	—	—	—	4 步	W0.04

② 功能图如图 6.19 所示。注意该系统是一种连续工作方式，即在初始点 ST1 处，按一下启动按钮，系统进行连续循环工作，直到按一下停止按钮，系统在完成最后一个周期的工作后，停止在 ST1 处。

0 步(W0.00 步)为等待步，W1.00 是连续工作状态位。W0.00 到 W0.01 的转换条件为 W1.00=1。

③ 设计梯形图

根据功能表图画出梯形图，如图 6.20 所示。

**例 6-2-4 波轮式洗衣机控制。**洗衣机洗衣筒工作示意图如图 6.21 所示。当筒内放好衣物和洗涤剂，并注满适量水以后，按下启动按钮，洗衣筒先正转 20 s，暂停 2 s，然后反转 20 s，再停 2 s，继续下一个循环。

① 系统的 I/O 分配见表(带步分配) 6.4。注意选用闭合开关控制输入点 0.00。

表 6.4 I/O 分配

输 入		输 出		步(0 步为等待步)	
闭合开关	0.00	正转接触器	100.00	0 步	W0.00
—	—	反转接触器	100.01	1 步	W0.01
—	—	—	—	2 步	W0.02
—	—	—	—	3 步	W0.03
—	—	—	—	4 步	W0.04

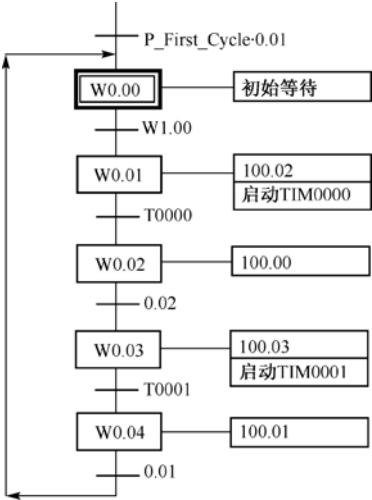


图 6.19 功能图

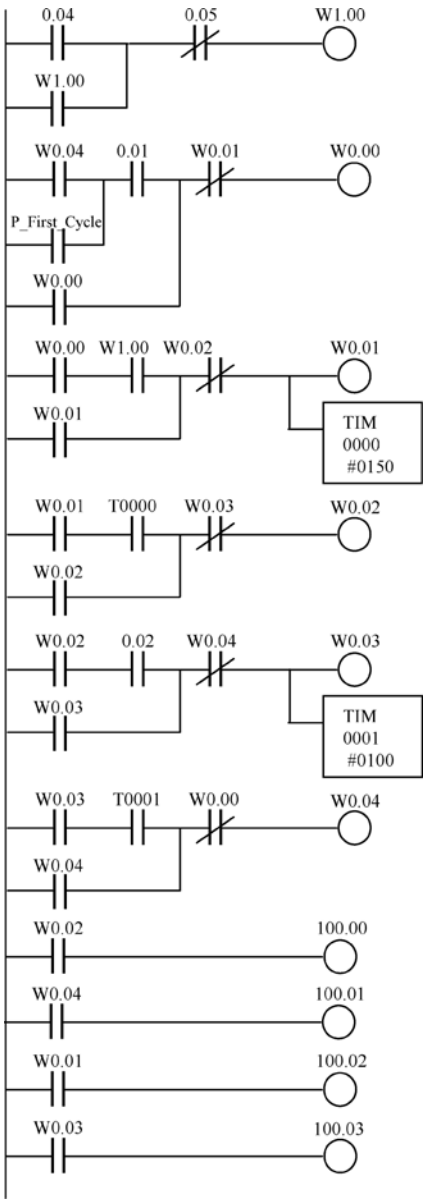


图 6.20 梯形图

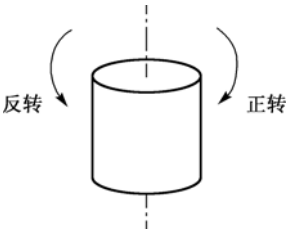


图 6.21 波轮式洗衣机洗衣筒工作示意图

- ② 顺序功能图如图6.22所示。
  - ③ 设计梯形图
- 根据功能表图画出梯形图，如图6.23所示。



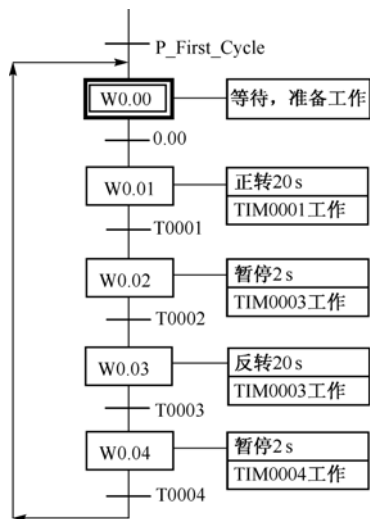


图 6.22 顺序功能图

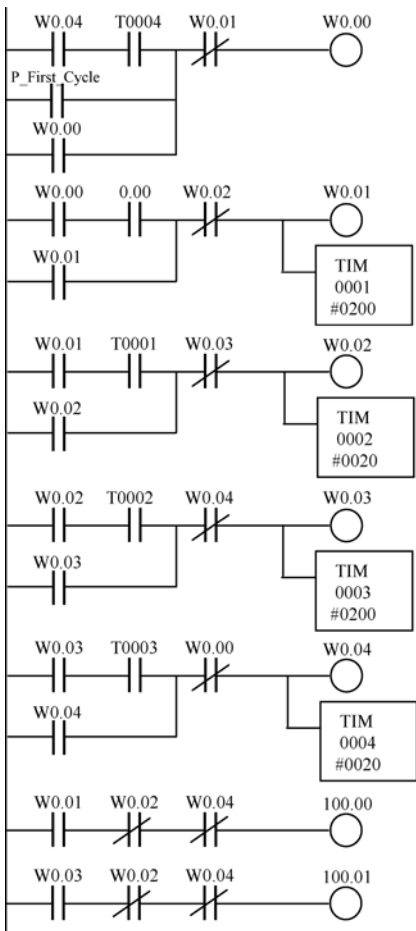


图 6.23 洗衣机梯形图程序

3. 用 SET 和 RSET 指令编写顺序控制程序

各种品牌的 PLC 都有一对置位 SET 指令与复位 RSET 指令。SET 指令可使操作位保持 ON 状态，直到该操作位被 RSET 指令复位为 OFF 止。SET 指令与 RSET 指令联合使用，能够严格实现顺序功能图编程的规则，如图6.24所示。

当图6.24中的 W0.00 为活动步，且转换条件 0.01 为 ON 时，SET 指令恰好将下一步 W0.01 置位变为活动步，而同时 RSET 指令将上一步 W0.00 复位变为不活动步。因此两个指令联合使用，可以实现顺序控制功能。

**例 6-2-5** 某加工自动线有一个钻孔动力头拟用 PLC 控制，其工作过程如图6.25所示。控制要求如下：

- ① 动力头在原位，按启动按钮，这时接通电磁阀 YV1，动力头快进。
- ② 动力头碰到行程开关 SQ1 后，接通电磁阀 YV1、YV2，动力头工进。
- ③ 动力头碰到行程开关 SQ2 后，电磁阀 YV1、YV2 断电，并开始延时。
- ④ 停留 15 s 后，接通电磁阀 YV3，动力头快退。
- ⑤ 动力头回到原位，碰到行程开关 SQ0 时自动停止，且停止指示灯亮。

试画出钻孔动力头的顺序功能流程图，并写出 PLC 控制程序。

解：从动力头工作示意图可见，是一个典型的顺序控制。

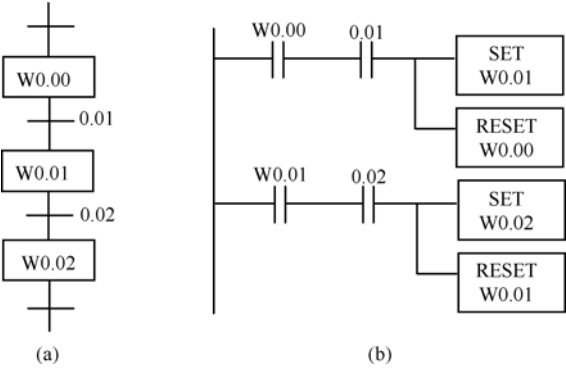


图 6.24 SET、RSET 指令的使用

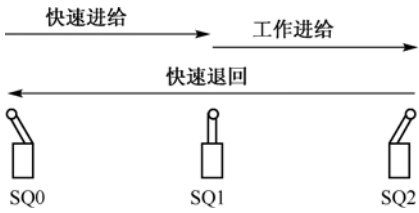


图 6.25 动力头工作过程示意图

① 输入/输出点分配：4 个输入点，4 个输出点，见表 6.5。

表 6.5 I/O 分配

输 入		输 出	
行程开关 SQ0	0.00	停止指示灯	100.00
行程开关 SQ1	0.01	电磁阀 YV1	100.01
行程开关 SQ2	0.02	电磁阀 YV2	100.02
启动按钮	0.05	电磁阀 YV3	100.03

② 顺序功能图：顺序功能图如图 6.26 所示，其中每步的含义见表 6.6。

③ 梯形图程序如图 6.27 所示。

表 6.6 步的含义

W0.00	W0.01	W0.02	W0.03	W0.04
等待步	快进	工进	停止 15 s	快退

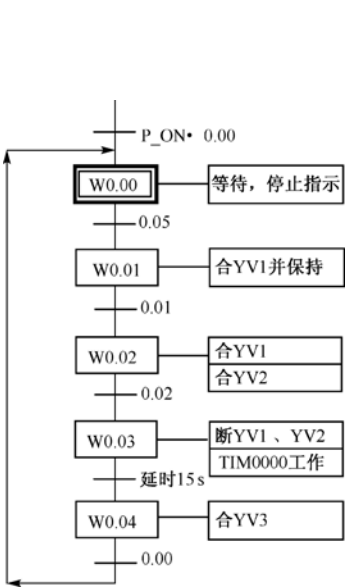


图 6.26 顺序功能图

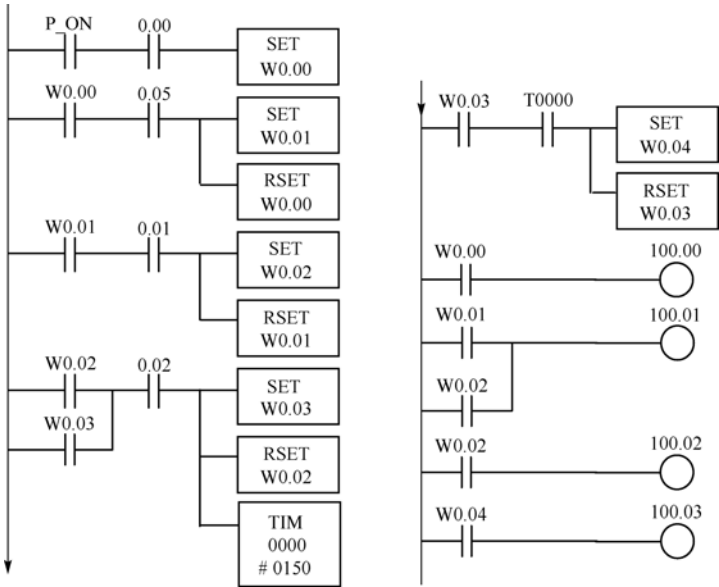


图 6.27 梯形图程序



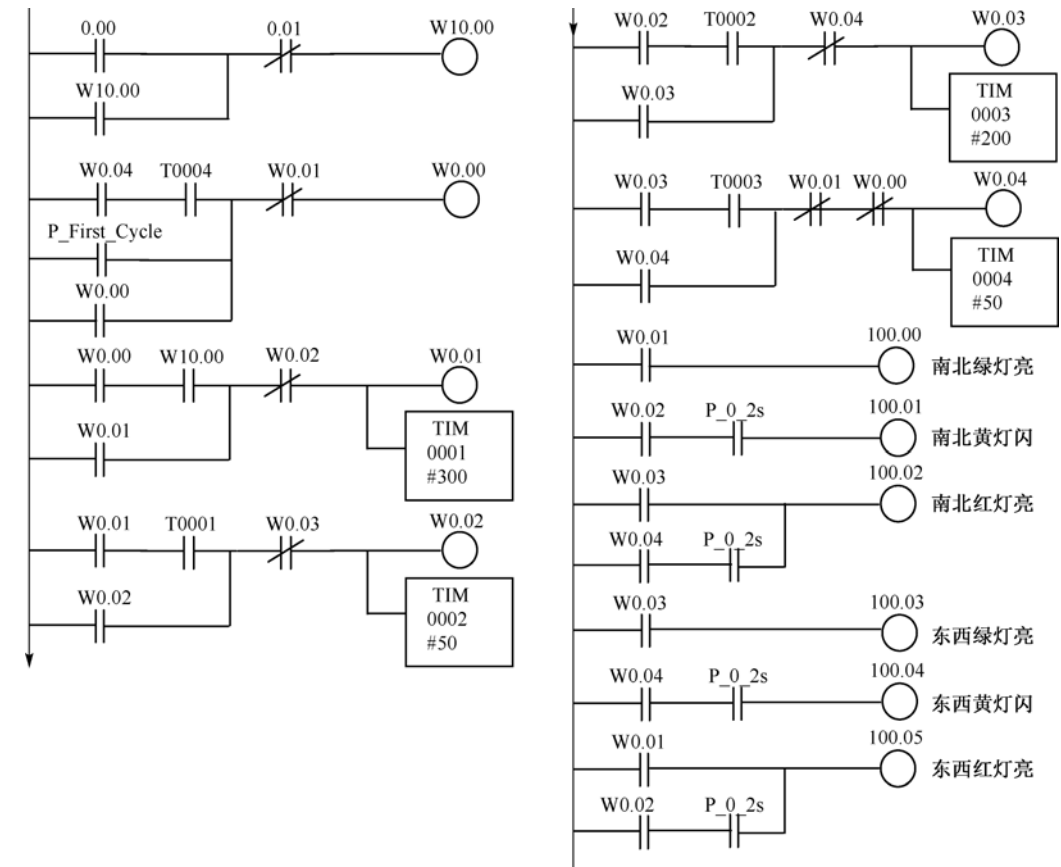


图 6.31 交通信号灯控制梯形图程序

6.2.4 经验设计法

在熟悉继电器控制电路设计方法的基础上，如果能透彻地理解 PLC 各种指令的功能，凭着经验能比较准确地选择使用 PLC 的各种指令而设计出相应的程序。这种方法没有固定模式可循，设计出的程序质量与个人的经验有很大关系。下面通过例子说明经验法的大体步骤。

- 例 6-2-7** 有一部电动运输小车供 8 个加工点使用。对小车的控制有以下几个要求：
- ① PLC 加电后，车停在某加工点(下称工位)，若没有用车呼叫(下称呼车)时，则各工位的指示灯亮，表示各工位可以呼车。
  - ② 若某工位呼车(按本位的呼车按钮)时，各位的指示灯均灭，表示此后再呼车无效。
  - ③ 停车位呼车则小车不动。当呼车位号大于停车位号时，小车自动向高位行驶，当呼车位号小于停车位号时，小车自动向低位行驶。当小车到达呼车位时自动停车。
  - ④ 小车到达某位时应停留 30 s 供该工位使用，不应立即被其他工位呼走。
  - ⑤ 临时停电后再复电，小车不会自行启动。
- 对本例的控制要求，可参照下面的步骤进行程序设计。
- ① 确定输入、输出电器。每个工位应设置一个限位开关、一个呼车按钮，系统要有用于启动和停机的按钮，这些是 PLC 的输入元件；小车要用一台电动机拖动，电动机正转时小车驶向高位、反转时小车驶向低位，电动机正转、反转各需要一个接触器，是 PLC 的执行元件。另外各工位还要有指示灯做呼车显示。电动机和指示灯是 PLC 的控制对象。

各工位的限位开关和呼车按钮的布置如图6.32所示，图中 ST 和 SB 的编号也是各工位的编号。

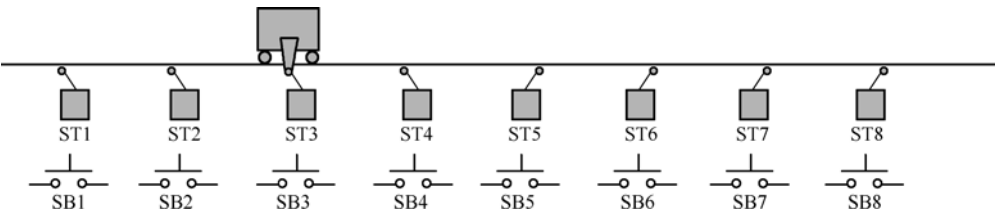


图 6.32 各加工位的限位开关、呼车按钮布置图

② 确定输入和输出点的个数，选择 PLC 机型，做出 I/O 分配。为了尽量减少占用 PLC I/O 点的个数，对本例，由于各工位的呼车指示灯状态一致，因此可选用小电流的发光元件并联在一起，然后接在一个 PLC 输出点上。使用 CP1 H 时所做的 I/O 分配见表 6.8。

表 6.8 I/O 分配

输 入				输 出	
限位开关 ST1	0.01	呼车按钮 SB1	1.01	呼车指示灯	101.07
限位开关 ST2	0.02	呼车按钮 SB2	1.02	电动机正转接触器线圈	100.00
限位开关 ST3	0.03	呼车按钮 SB3	1.03	电动机反转接触器线圈	100.01
限位开关 ST4	0.04	呼车按钮 SB4	1.04	—	—
限位开关 ST5	0.05	呼车按钮 SB5	1.05	—	—
限位开关 ST6	0.06	呼车按钮 SB6	1.06	—	—
限位开关 ST7	0.07	呼车按钮 SB7	1.07	—	—
限位开关 ST8	0.08	呼车按钮 SB8	1.08	—	—
系统启动按钮	0.00	—	—	—	—
系统停止按钮	0.10	—	—	—	—

③ 为了分析问题方便，可先画出系统动作过程的流程图，如图6.33所示。

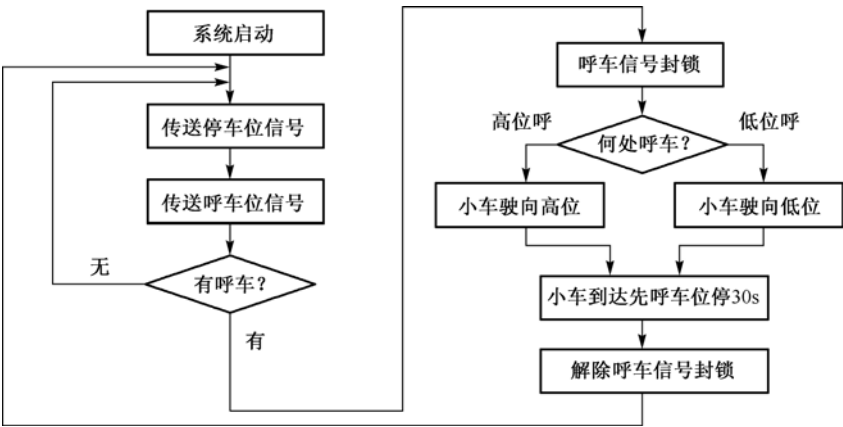


图 6.33 运输小车系统流程图

④ 选择 PLC 指令并编写程序。选择指令是一个经验问题。对于本例的控制要求，一般会想到用 MOV 指令和 CMP 指令，即先把小车所在的工位号传送到一个通道中，再把呼车的工位号传送到另一个通道中，然后将这两个通道的内容进行比较。若呼车的位号大于停车的位号，则小车向高位行驶；若呼车的位号小于停车的位号，则小车向低位行驶。对小车的这种控制，是本例程序设计的主线。

⑤ 编写其他控制要求的程序。其一，若有某位呼车则应立即锁定其他位的呼车信号；其二，小

车行驶到位后应在该位停留一段时间，即延迟一定时间再解除对呼车信号的锁定；其三，失压保护程序；其四，呼车显示程序。

⑥ 将对各环节编写的程序合理地联系起来，即得到一个满足控制要求的程序。本例设计的程序如图6.34所示。

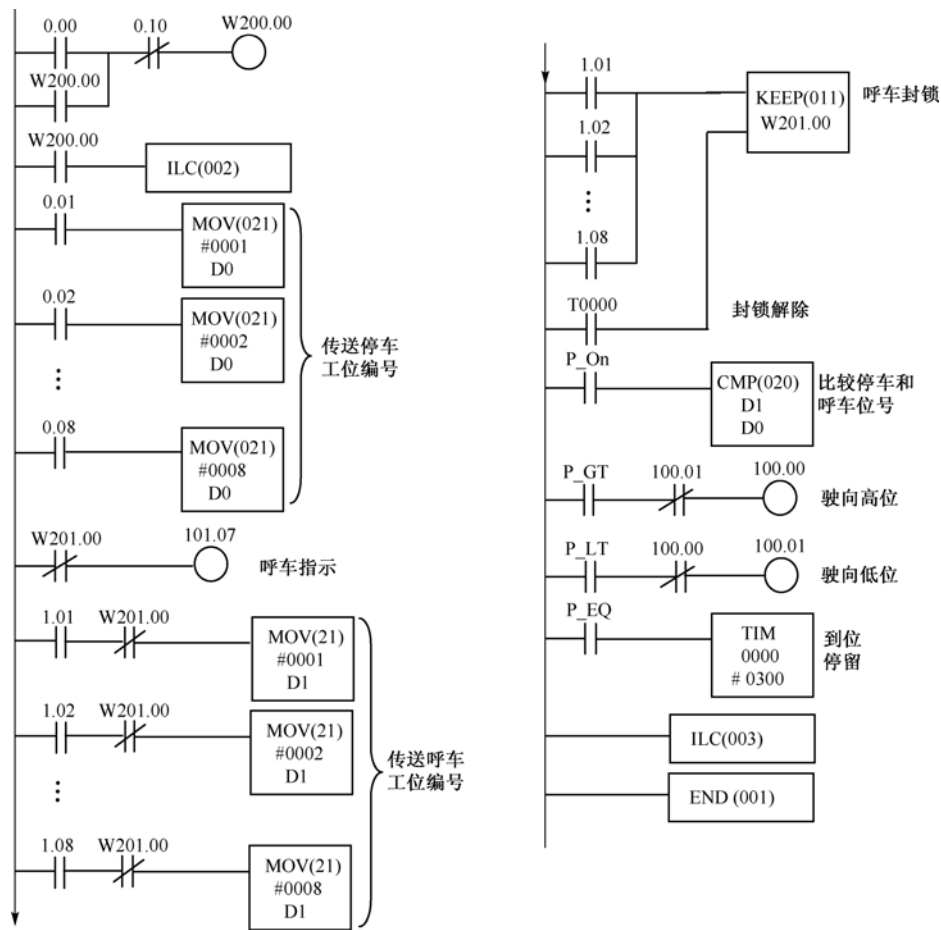


图 6.34 运输小车自动控制程序

对图6.34程序的设计意图和控制功能简要分析如下。

- ① 用 MOV 指令分别向 D0 通道传送车位信号、向 D1 通道传送各位的呼车信号。没有呼车时，W201.00 为 OFF、101.07 为 ON，各位的指示灯亮，示意各工位可以呼车。
- ② 本例用 KEEP 指令进行呼车锁定和解除锁定的控制。只要某位呼车，就执行 KEEP 指令，将 W201.00 置为 ON，从而使其他传送呼车信号的 MOV 指令不能执行，实现先呼车的位优先用车。同时指示灯灭，示意别的位不能呼车，即呼车锁定开始。
- ③ 执行 CMP 指令可以判别呼车位号比停车车位号大还是小，从而决定小车的行驶方向。若呼车位号比停车车位号大，则 100.00 为 ON，小车驶向高位。在行车途中经由各位时必然要压动各位的限位开关，即行车途中 00 通道的内容随时改变，但由于其位号都比呼车位号小 (01 通道中的呼车位号不变)，故可继续行驶直至到达呼车位。若呼车位号比停车车位号小，则小车驶向低位。在行车途中要压动各位的限位开关，但其位号都比呼车位号大，故可继续行驶直至到达呼车位。
- ④ 当小车到达呼车位时，其一使 P\_GT 或 P\_LT 变为 OFF，使 100.00 或 100.01 为 OFF，小车停

在呼车位；其二使 P\_EQ 变为 ON，则立即启动 TIM0000 开始定时，使小车在呼车位停留 30 s。30 s 到，使 W201.00 复位，指示灯亮并解除呼车锁定。此后各工位又可以开始呼车。

⑤ 若系统运行过程中掉电再复电时，不按启动按钮程序是不会执行的。另外，在 PLC 外部也设置失压保护措施，所以掉电再复电时，小车不会自行启动。

**例 6-2-8** 保留例 6-2-7 的全部要求，但把第 4 个控制要求修改为：给位号高的加工位以优先用车的机会，8 号位优先权最高。

图 6.34 的程序不再能胜任本例的控制要求。欲区别呼车位的位号大小，如果使用比较指令将会使程序非常烦琐。但使用编码、译码指令，程序就会简练得多。因为编码指令只对编码通道的最高位进行编码，因此使用编码指令编程时，能实现给高位号的工位获得优先用车的机会。

编写这个程序的思路是，在呼车锁定解除的时间内，用编码指令随时对呼车信号通道 01 进行编码。假定几个工位都按住呼车按钮不放，一直按到下一次呼车锁定（看到呼车指示灯灭），则高位号的工位就可以优先用车了。编码之后再进行译码，把译码结果通道 W201 的内容与停车位信号通道 00 的内容进行比较，就可以决定小车的行驶方向。

鉴于上述构思，系统功能流程图如图 6.35 所示。

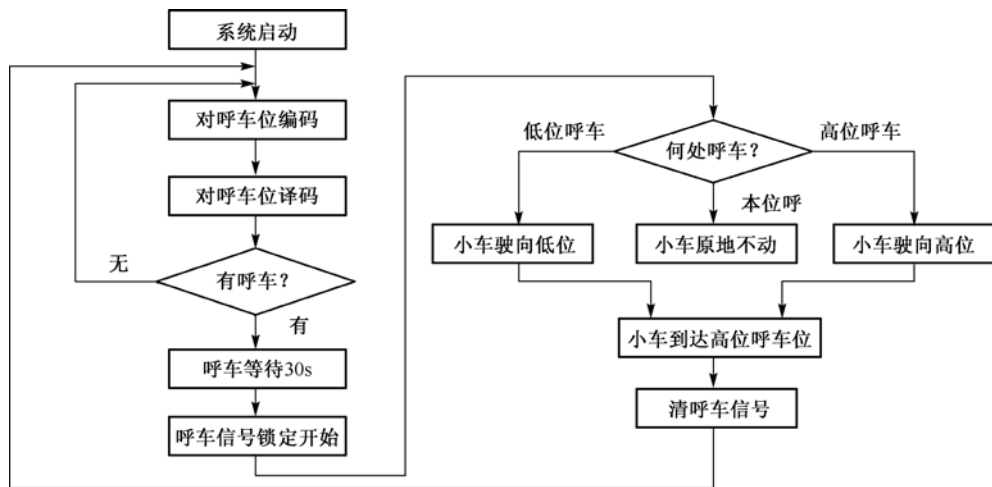


图 6.35 系统功能流程图

根据流程图和 I/O 分配(同例 6-2-7)设计的控制程序如图 6.36 所示。对图 6.36 的控制功能简要分析如下：

① PLC 加电后按一下启动按钮，程序即开始执行。由于小车停在某位没有启动，100.00 和 100.01 为 OFF，所以呼车指示灯亮示意各位可以呼车。

② 执行 DMPX 指令对呼车信号 01 通道进行编码，编码结果放在 W200 中；执行 MLPX 指令对 W200 通道进行译码，译码结果放在 W201 中；执行 CMP 指令把通道 01 的内容与 #0000 进行比较以查看是否有呼车，若有则进行下一步，若无则等待呼车。

③ 如果有呼车则 P\_GT 为 ON，使 W202.00 ON 并保持。随之执行下一个 CMP 指令，把通道 00(停车位号)与 W201(呼车位号)的内容进行比较。当呼车的位号大于或小于停车位号时，启动定时器开始定时 30 s(呼车等待时间)。此间指示灯仍亮，各位仍可呼车、即 W201 通道中的内容可变。

④ 当呼车等待时间到，定时完成标志 T0000 为 ON 时，若最后一次呼车的位号大于停车的位号则 100.00 为 ON，小车驶向高位；若最后一次呼车的位号小于停车的位号则 100.01 为 ON，小车驶向

低位。不论 100.00 还是 100.01 为 ON，其作用都有二：其一，使 101.07 为 OFF、指示灯灭，示意各位不能再呼车，即开始呼车锁定；其二，编码、译码指令都停止执行，即此后一直到锁定解除的时间内，W201 通道中的内容不变。

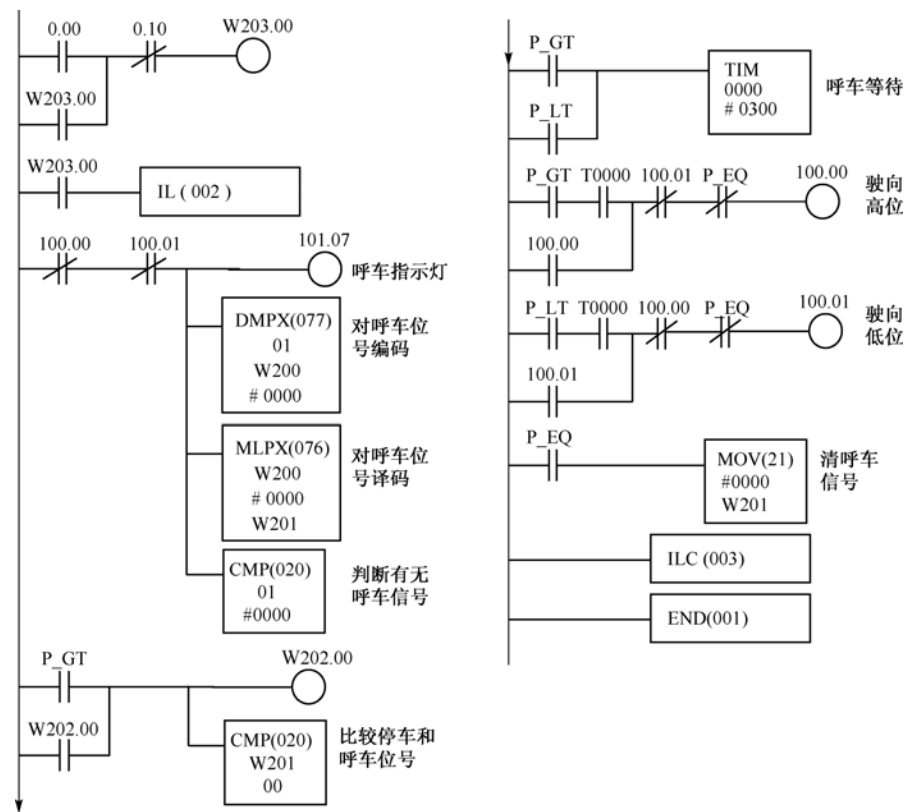


图 6.36 例 6-2-8 小车控制梯形图

⑤ 若小车驶向高位，在行车途中经由各位，必然要压动各位的限位开关(00 通道的内容可改变)，但其位号都比呼车位号小(W201 通道中呼车位号不变)，所以可继续前进直至达到呼车位。若小车驶向低位，在行车途中压动各位的限位开关，但其位号都比呼车位号大，所以可继续前进直至达到呼车位。

⑥ 当小车到达呼车位时 P\_EQ 为 ON，其作用有四：其一，使 100.00 或 100.01 为 OFF，则小车停在呼车位；其二，执行 MOV 指令将呼车信号通道 W201 清零；其三，101.07 为 ON、指示灯亮，示意各位可以呼车；其四，又开始对呼车位号进行编码和译码。

例 6-2-7 和例 6-2-8 的呼车方式有所不同。例 6-2-7 是先呼车者优先用车，例 6-2-8 则分两种情况区别对待：其一，在锁定解除的时间内，相继呼车的各工位后呼车者先用车。因为最后一次执行编码指令时只能对最后呼车的工位进行编码；其二，呼车锁定解除时间到(指示灯灭的瞬时)，同时呼车的各工位，位号大的先用车。这样，若急于用车的工位可按住呼车按钮不放。

6.2.5 继电器控制电路图转换设计法

用 PLC 控制的系统或设备，功能完善，可靠性好，所以用 PLC 控制取代继电器控制已是大势所趋。有些继电器控制的系统或设备，经过多年的运行实践证明其设计是成功的，若欲改用 PLC 控制，可以在原继电器控制电路的基础上，经过合理转换，或者说经适当“翻译”，从而设计出具有相同功能的 PLC 控制程序。把继电器控制转换成 PLC 控制时，要注意转换方法，以确保转换后系统的功能不变。



1. 对各种继电器、电磁阀等的处理

在继电器控制的系统中，大量使用各种继电器，如交或直流接触器、电磁阀、电磁铁、中间继电器等。交或直流接触器、电磁阀、电磁铁的线圈是执行元件，要为它们分配相应的 PLC 输出继电器号，中间继电器可以用 PLC 内部的继电器来代替。

2. 对常开、常闭按钮的处理

在继电器控制电路中，一般启动用常开按钮，停车用常闭按钮。用 PLC 控制时，启动和停车一般都用常开按钮。尽管使用哪种按钮都可以，但是画出的 PLC 梯形图却不同。

图6.37 中，SB1 是启动按钮、SB2 是停车按钮，KM 是交流接触器。图6.37(a)的停车用常开按钮，对应梯形图中的 0.01 是常闭触点；图6.37(b)中的停车用常闭按钮，对应梯形图中的 0.01 是常开。在转换时这一点要注意。

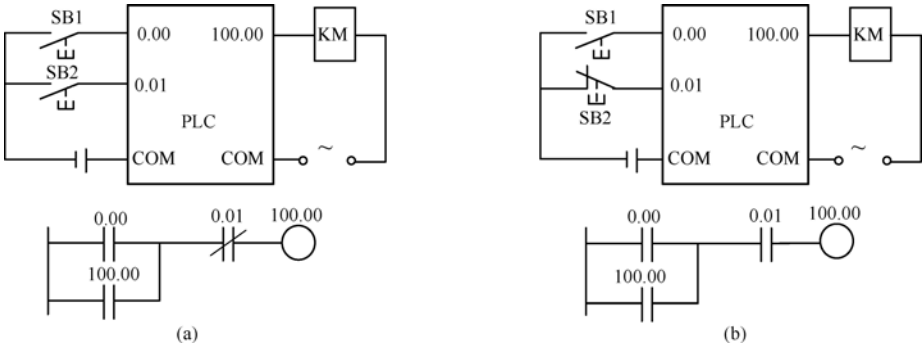


图 6.37 按钮与梯形图的对应关系

3. 对热继电器触点的处理

若 PLC 的输入点较富裕，热继电器的常闭触点可占用 PLC 的输入点，若输入点较紧张，热继电器的信号可不输入 PLC 中，而接在 PLC 外部的控制电路中。

4. 对时间继电器的处理

物理的时间继电器可分为通电延时型和断电延时型两种。通电延时型时间继电器，其延时动作的触点有通电延时闭合和通电延时断开两种。断电延时型时间继电器，其延时动作的触点有断电延时闭合和断电延时断开两种。用 PLC 控制时，时间继电器可以用 PLC 的定时器/计数器来代替。PLC 定时器的触点只有接通延时闭合和接通延时断开两种，但通过编程，可以设计出满足要求的时间控制程序。

对于图6.38(a)的一段控制电路，时间继电器是通电延时型的。当过流继电器 KA 的常开触点接通时，时间继电器开始定时，延时后 KM 线圈得电，该图对 KM 实现了延时接通的控制。对图6.38(a)中的各电器做 I/O 分配：KA 对应 PLC 输入点为 0.00，KM 对应输出点为 100.00。KT 用 TIM0000 代替。画出 PLC 的梯形图如图6.38(b)所示。

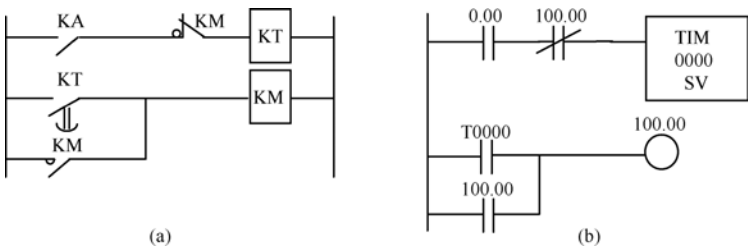


图 6.38 通电延时接通的控制

5. 处理电路的连接顺序

在转换成 PLC 的梯形图时，一般要把控制电路图做一点调整，这样能方便转换。

例如，将图6.39(a)转换成 PLC 梯形图时，要先对图6.39(a)中电路图的部分接线进行调整。线圈 KM2 和 K 之间连接着常开触点 KM2，PLC 的梯形图不允许有这种结构，对这种接线图要进行调整。由于 K 接通的条件有两个，其一是 KM2 接通，其二是时间继电器的常开触点 KT 闭合，两者具一即可。所以，应将 KM2 的常开触点与 KT 的延时闭合的常开触点并联作为 K 的接通条件。根据这个原则，画出调整后的控制电路如图6.39(b)所示。对图6.39(b)的电路做 I/O 分配如图6.39(c)所示，中间继电器用 W200.00，时间继电器用 TIM0000 来代替。由 I/O 分配画出 PLC 的梯形图如图6.39(d)所示。

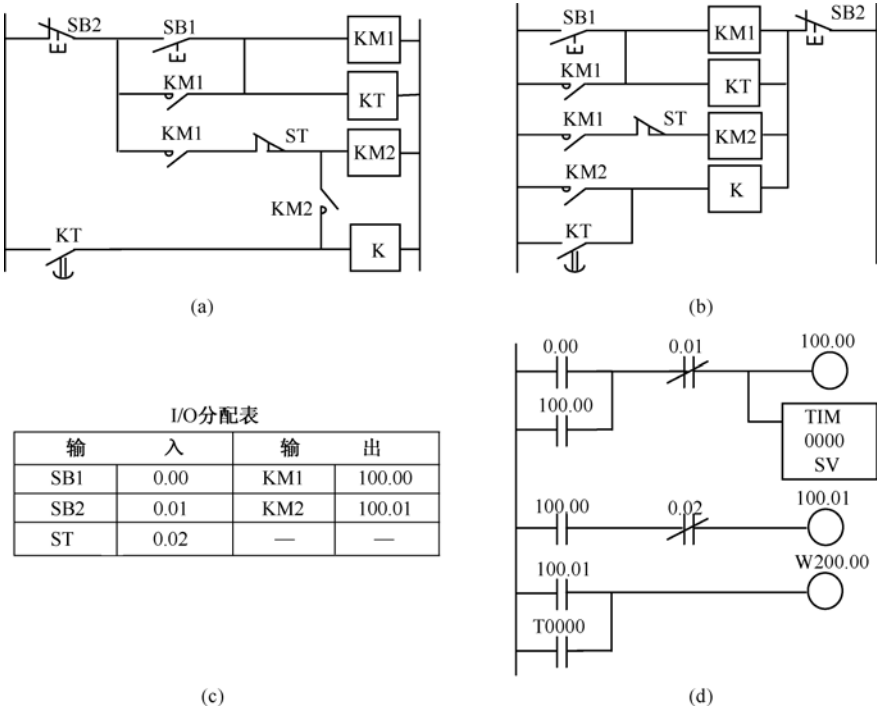


图 6.39 控制电路图接线的调整

由继电器控制电路转换成 PLC 梯形图后，一定要仔细校对、认真调试，以保证其控制功能与原图相符。

本节所举的例子只是控制电路的局部。对复杂的控制电路可以化整为零，先进行局部的转换，最后再综合起来。当控制电路很复杂时，大量的中间继电器、时间继电器、计数器等，都可以用 PLC 的内部器件来取代，复杂的控制逻辑可用程序来实现，这时，用 PLC 取代继电器控制的优越性就显而易见了。

6.2.6 具有多种工作方式系统的编程方法

不少系统需要具备多种工作方式，例如，既能自动地循环运行一个过程，也能进行手动操作运行一个工作步等。常见的工作方式有连续、单周期、单步和手动。所谓连续方式是指系统启动后连续、周期性地运行一个过程，单周期方式是指启动一次只运行一个工作周期，单步方式是启动一次只能运行一个工作步，手动方式与点动控制相似。

对一个设备来说，几种工作方式不能同时运行。所以在设计这类程序时，可以对几种工作方式的

程序分别进行处理,最后综合起来,这样可以简化程序的设计。下面通过一个例子说明多种工作方式的程序设计问题。在本例中,还提出了编写误操作禁止程序的实际问题。

**例 6-2-9** 采用液压控制的搬运机械手,其任务是把左工位的工件搬运到右工位,图6.40是其动作示意图。机械手的工作方式分为手动、单步、单周期和连续4种。机械手各种工作方式的动作过程及控制要求叙述如下。

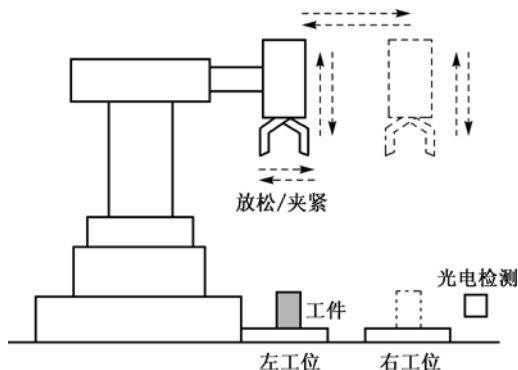


图 6.40 机械手动作示意图

### 1. 机械手的工作方式

#### (1) 单周期方式

机械手在原位压左限位开关和上限位开关。按一次操作按钮机械手开始下降,下降到左工位压动下限位开关后自停;接着机械手夹紧工件后开始上升,上升到原位压动上限位开关后自停;接着机械手开始右行直至压动右限位开关后自停;接着机械手下降,下降到右工位压动下限位开关(两个工位用一个下限位开关)后自停;接着机械手放松工件后开始上升直至压动上限位开关后自停(两个工位用一个上限位开关);接着机械手开始左行直至压动左限位开关后自停。至此一个周期的动作结束,再按一次操作按钮则开始下一个周期的运行。

#### (2) 连续方式

启动后机械手反复运行上述每个周期的动作过程,即周期性连续运行。

#### (3) 单步方式

每按一次操作按钮,机械手完成一个工作步。例如,按一次操作按钮机械手开始下降,下到左工位压动下限位开关自停,欲使之运行下一个工作步,必须再按一次操作按钮等。

以上3种工作方式属于自动控制方式。

#### (4) 手动方式

按下按钮则机械手开始一个动作,松开按钮则停止该动作。

### 2. 对机械手每个工作步的控制要求

#### (1) 上升和下降

机械手上升或下降的动作都要到位,否则不能进行下一个工作步。本例使用上、下限位开关进行控制。上升/下降的动作用一个双线圈的电磁阀控制。

#### (2) 夹紧和放松

机械手夹紧和放松的动作必须在两个下工位处进行,且夹紧和放松的动作都要到位。

为了确保夹紧和放松动作的可靠性,本例对夹紧和放松动作进行定时,并设置夹紧和放松指示。夹紧和放松动作由单线圈的电磁阀控制。

#### (3) 左行和右行

自动方式时,机械手的左、右运动必须在压动上限位开关后才能进行;机械手的左、右运动都必须到位,以确保在左工位取到工件,在右工位放下工件。本例利用上限位、左限位、右限位开关进行控制。左/右行的动作由双线圈的电磁阀控制。

### 3. 自动方式下误操作的禁止

自动方式(连续、单周期、单步)时,按一次操作按钮自动运行方式开始后,此后再按操作按钮属错误操作,程序对错误操作不予响应。

另外，当机械手到达右工位上方时，下一个工作步就是下降。为了确保在右工位没有工件时才能开始下降，所以应在右工位设置有无工件检测装置。本例使用的是光电检测装置。

根据上述控制要求，操作盘上要设置一个 PLC 的电源开关(不占输入点)、一个工作方式选择开关和一个动作选择开关，通过这两个开关选择工作方式和动作方式、操作按钮和停车/复位按钮各一个，这两个按钮其他作用见操作盘面板。操作盘面板的布置如图 6.41 所示。

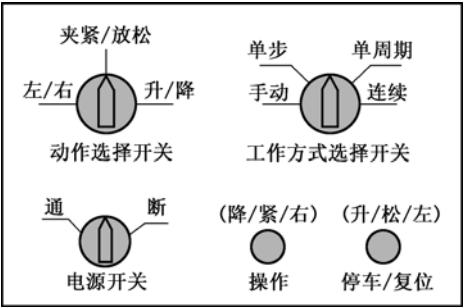


图 6.41 操作盘面板的布置

4. I/O 分配

依据控制要求，需要 14 个输入点、8 个输出点，选用 CP1H 的 I/O 分配见表 6.9。

表 6.9 I/O 分配

输 入				输 出	
操作按钮	0.00	升降选择	1.00	下降电磁阀线圈	100.00
停车按钮	0.01	紧/松选择	1.01	上升电磁阀线圈	100.01
下降限位	0.03	左/右选择	1.02	紧/松电磁阀线圈	100.02
上升限位	0.04	手动方式	1.03	右行电磁阀线圈	100.03
右行限位	0.05	单步方式	1.04	左行电磁阀线圈	100.04
左行限位	0.06	单周方式	1.05	原位指示灯	100.05
光电开关	0.07	连续方式	1.06	夹紧指示灯	100.06
—	—	—	—	放松指示灯	100.07

5. 程序设计

在进行程序设计之前，先画出机械手的动作流程图，如图 6.42 所示。流程图中，能清楚地看到机械手每一步的动作内容及步间的转换关系。

根据流程图，设计出应用程序的总体方案，如图 6.43 所示。图中，把整个程序分为两大块，即手动和自动两部分。当选择开关拨到手动方式时，输入点 1.03 为 ON，其常开触点接通，开始执行手动程序；当选择开关拨在单步、单周期或连续方式时，输入点 1.03 断开，其常闭触点闭合，开始执行自动程序。至于执行自动方式的哪一种，则取决于方式选择开关是拨在单步、单周期还是连续的位置上。

(1) 手动程序设计

图 6.44 是根据要求设计的手动控制程序的梯形图，对其功能做如下分析。

① 上升/下降控制(工作方式选择开关拨在手动位)。

手动控制机械手的升/降、左/右行、工件的夹紧/放松操作，是通过方式开关、操作和停车按钮的配合来完成的。

欲进行机械手升降操作时，要把选择开关拨在升/降位，使 1.00 接通。下降操作为，按下操作按钮时输入点 0.00 接通，则 100.00(下降电磁阀线圈)接通使机械手下降，松开按钮则机械手停。当按住操作按钮不放时，机械手下降到位压动下限位开关 0.03 时自停。

上升操作为，按下停车按钮时输入点 0.01 接通，则 100.01(上升电磁阀线圈)接通使机械手上升，松开按钮时机械手停。当按住停车按钮不放时，机械手上升到位压动上限位开关 0.04 后自停。

② 夹紧/放松控制(工作方式选择开关拨在手动位)。

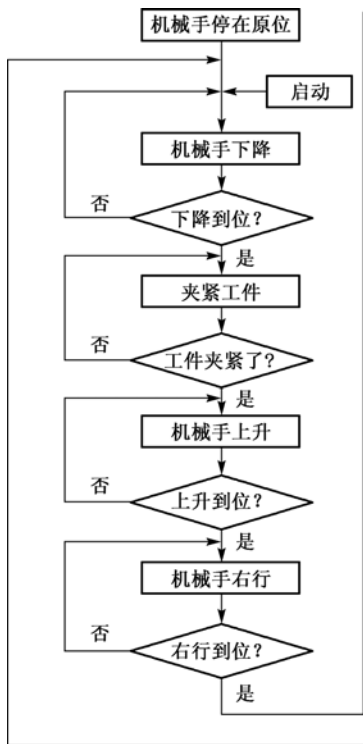


图 6.42 机械手自动运行流程图

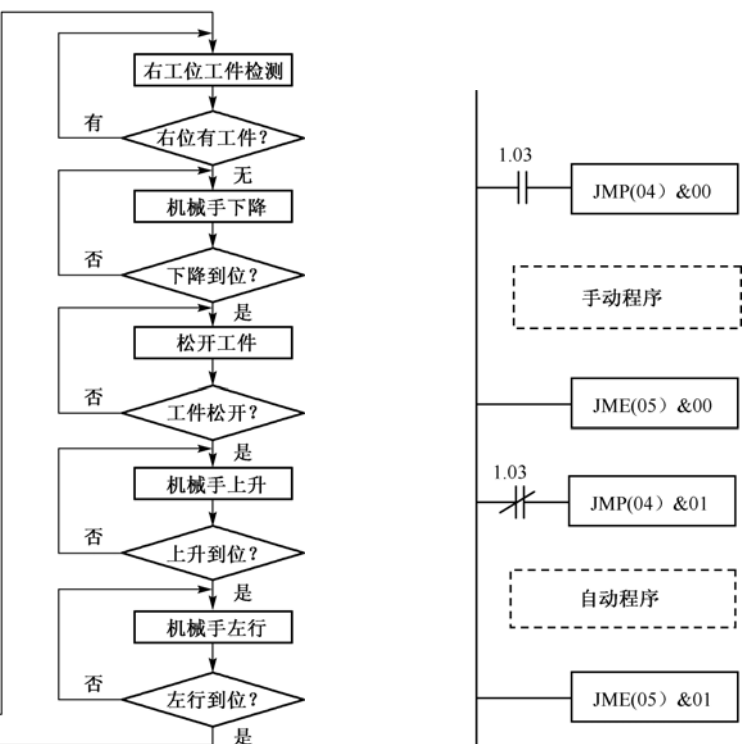


图 6.43 程序总体方案

只有机械手停在左或右工作位，下限位开关 0.03 受压(其常开触点接通)时，夹紧/放松的操作才能进行。要把动作选择开关拨在夹紧/放松位，使输入点 1.01 接通。

若机械手停在左工作位，压下限位开关，且此时有工件时，当按住操作按钮时如下动作开始：其一，100.02 被置位，机械手开始夹紧工件；其二，100.06 为 ON，夹紧动作指示灯亮，表示正在进行夹紧的动作；其三，TIM0002 开始夹紧定时。当定时时间到，夹紧动作指示灯灭时，方可以松开按钮。此时 100.02 仍保持接通状态，TIM0002 被复位。

若机械手停在右工作位，压下限位开关，且夹有工件时，当按住停车按钮时如下动作开始：其一，100.02 被复位，机械手开始放松工件；其二，100.07 为 ON 使放松动作指示灯亮，表示正在进行放松的动作；其三，TIM0003 开始放松定时。当定时时间到，放松动作指示灯灭时，方可以松开按钮，此时 100.02 仍保持断开状态，TIM0003 复位。

③ 左行/右行控制(工作方式选择开关拨在手动位)。  
把动作选择开关拨在左/右位，使输入点 1.02 接通。

右行的操作为，按住操作按钮 0.00，100.03(右行电磁阀线圈)得电使机械手右行，松开按钮则机械手停。当按住操作按钮不放时，机械手右行，右行到位压动右限位开关 0.05 时自停。  
左行的操作为，按住停车按钮 0.01，100.04(左行电磁阀线圈)得电使机械手左行，松开按钮则机械手停。当按住停车按钮不放时，机械手左行，左行到位压动左限位开关 0.06 时自停。

(2) 自动程序设计

图6.45是根据要求设计的自动控制程序的梯形图，对其功能做如下分析。

① 连续运行方式的控制(工作方式选择开关拨在连续位)。  
连续运行方式的启动必须从原位开始。如果机械手没停在原位，要用手动操作让机械手返回原位。当机械手返回原位时，原位指示灯亮。

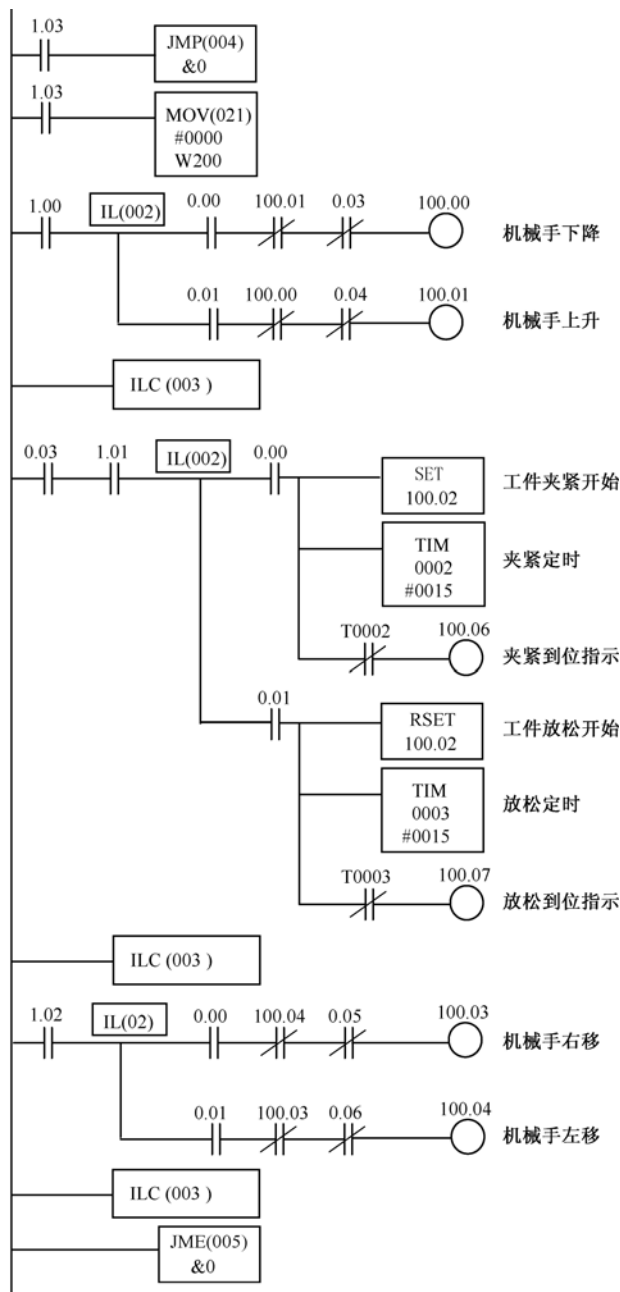


图 6.44 机械手的手动控制梯形图

方式选择开关拨在连续位且输入点 1.06 接通，其一使 W210.00 置位，其二使 SFT 的移位脉冲输入端接通。

移位寄存器通道 W200，是由 P\_First\_Cycle 或停止按钮 0.01 进行复位的。

由于机械手在原位，上限位开关和左限位开关受压，常开触点 0.04 和 0.06 都闭合。所以按一下操作按钮，则向移位寄存器发出第一个移位脉冲。第一次移位使 W200.00 为“1”，从而使 100.00 为 ON，自此机械手开始下降，且 0.04 和 0.06 均变为 OFF。

当机械手下降到左工位，压动下限位开关时，0.03 的常开触点闭合，于是移位寄存器移位一次。

由于机械手离开了原位，串联在移位输入端的常开触点 0.00、0.04 和 0.06 都是断开的，所以这次移位使 W200.00 变为“0”，而 W200.01 为“1”。

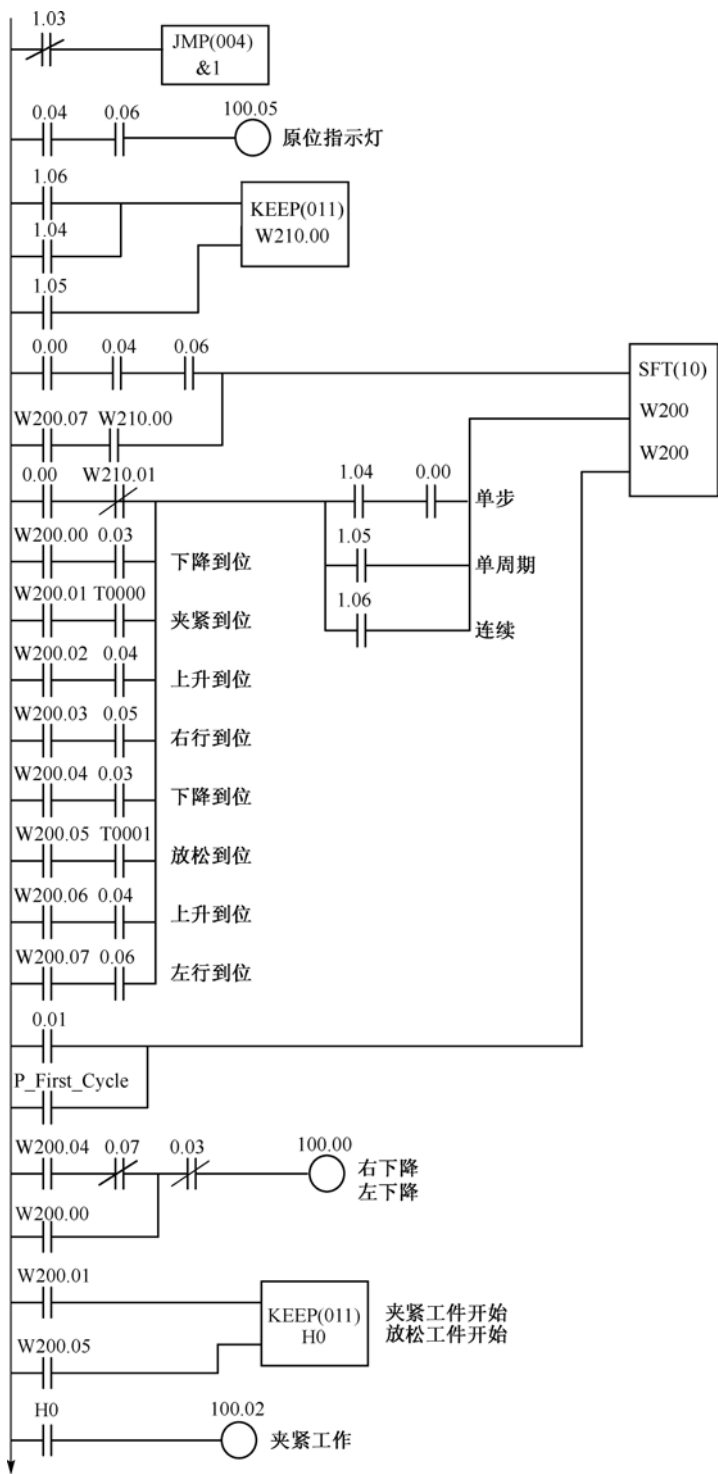


图 6.45 机械手自动控制梯形图

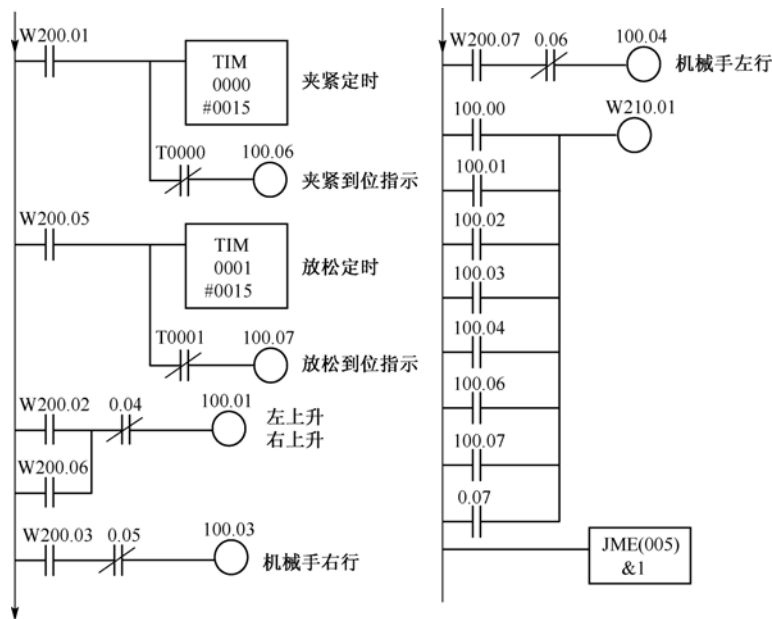


图 6.45 机械手自动控制梯形图(续)

W200.01 为“1”的作用是，其一使 H0 置位，100.02 为 ON，工件夹紧动作开始；其二使夹紧动作指示灯亮；其三使夹紧定时器 TIM0000 开始定时。当定时时间到，即夹紧到位时，夹紧指示灯灭，而移位寄存器又移位一次，使 W200.01 变为“0”，而 W200.02 变为“1”。

W200.02 为“1”使 100.01 为 ON，自此机械手开始上升。当机械手上升到原位时压上限位开关 0.04，使 100.01 断电，上升动作停止，同时移位寄存器又移位一次，使 W200.02 变为“0”，而 W200.03 变为“1”。

W200.03 为“1”使 100.03 为 ON，自此机械手开始右移。当机械手右移到位压右限位开关 0.05 时，使 100.03 断电、右移停止，同时移位寄存器又移位一次，使 W200.03 变为“0”，而 W200.04 变为“1”。

W200.04 为“1”时，若检测到右工位没有工件，光电开关的常闭触点 0.07 接通时，使 100.00 再次为 ON，自此机械手开始下降。当机械手下降到右工位压动下限位开关 0.03 时，100.00 断电，下降动作停止，同时移位寄存器又移位一次，使 W200.04 变为“0”，而 W200.05 变为“1”。若检测到右工位有工件，常闭触点 0.07 断开时，则机械手停在右上方不动。只有拿掉右工位的工件，机械手才开始下降。

W200.05 为“1”的作用是，其一，使 H0、100.02 复位，工件放松动作开始；其二，使放松动作指示灯亮；其三，放松定时器 TIM0001 开始定时。当定时时间到，即放松到位时，放松指示灯灭，而移位寄存器又移位一次，使 W200.05 变为“0”，W200.06 变为“1”。

W200.06 为“1”使 100.01 再次为 ON，自此机械手开始上升。当机械手上升至压动上限位开关 0.04 时，100.01 断电，上升动作停止，同时移位寄存器又移位一次，使 W200.06 变为“0”，而 W200.07 变为“1”。

W200.07 为“1”使 100.04 为 ON，自此机械手开始左移。当机械手左移到位时压左限位开关 0.06，100.04 断电，左移停止，移位寄存器又移位一次。由于 W200.07 和 W210.00 一直为 ON，所以 SFT 的数据输入端为“1”。这样，本次移位使 W200.00 又变为“1”，随之开始了下一个周期的运行。

② 单周期运行方式的控制(方式选择开关拨在单周期位)。

由于方式选择开关拨在单周期位时使 1.05 接通，其常开触点闭合使 W210.00 被复位。所以当机械手运行到一个循环的最后一步结束，W200.07 和左限位 0.06 为 ON 时，因 W210.00 已断开而使 SFT 的数据输入为“0”，不能使 W200.00 再置位，因此只能在一个周期结束时停止运行。要想进行下一个周期的运行，必须再按一次操作按钮。



### ③ 单步运行方式的控制(方式选择开关拨在单步位)。

单步方式时, SFT 的移位输入端是常开触点 1.04 与 0.00 的串联, 所以按一次操作按钮发一个移位脉冲, 机械手只完成一步的动作就停止。例如, 当 W200.00 接通机械手下降到位时, 0.03 被接通, 但此时若不再按一下操作按钮, 则移位信号不能送到 SFT 的移位输入端, 因此机械手只能在一步结束时停止运行。

由于方式选择开关拨在单步位, 1.04 接通, 其常开触点闭合, 使 W210.00 被置位。当机械手运行到一个循环的最后一步结束, 即 W200.07 和 0.06 为 ON 时, 由于移位输入端的 W200.07 和 W210.00 接通, 所以若再按一次操作按钮能使 W200.00 再置位, 即进入下一个周期的第一步。

### ④ 自动方式下误操作的禁止。

连续、单周期、单步都属于自动方式的运行。为了防止误操作, 本例编写了相应的程序段, 其原理是, 在自动运行过程中, 由于 100.00~100.07(除 100.05)及 0.07 中总有一个为 ON, 使 W210.01 总为 ON。由于常开触点 0.00 和常闭触点 W210.01 串联在移位寄存器的移位脉冲输入端, 这样, 在自动方式第一次按启动按钮自动运行开始后, 如果随后又误按了一下启动按钮 0.00, 程序不会响应。这是因为第一次按启动按钮后, 100.00 即 ON, 且使 W210.01 为 ON, 其常闭触点 W210.01 即断开, 此后再按启动按钮, 移位脉冲也不会送达 SFT 的 CP 端。同样其他各步也能保证 W210.01 为 ON, 所以启动后误按操作按钮不会造成误动作。

在使用移位寄存器时, 如果移位脉冲是通过操作按钮输入的, 都要考虑误操作的问题。因为误按操作按钮是难免的, 这个问题没处理好, 容易发生失控现象。

### ⑤ 手动和自动方式转换时的复位问题。

由于手动和自动的切换是由 JMP/JME 指令实现的, 当 JMP 的执行条件由 ON 变为 OFF 时, JMP 与 JME 之间的各输出状态保持不变。所以在手动方式与自动方式切换时, 一般要进行复位操作, 以避免出现错误动作。

由于自动运行方式必须是机械手停在原位时才能启动, 所以经过手动复位后, 使 100.00~100.07(除 100.05)都被复位。在自动运行过程中欲停机, 应按一次停车按钮 0.01 对 W200 通道进行复位, 也间接地对 100 通道复了位。

在自动运行过程中, 若未按停车按钮直接将方式开关(1.03)拨到手动位时, W200 通道中的状态将保持。当手动操作完毕再转到自动状态时, W200 通道的原状态就会导致误动作。为了防止这种现象发生, 在手动控制程序中采取了复位措施。由于 W200 通道被复位, 因此切换时不会出现误动作。

## 6.3 应用程序实例

前面各节已对 PLC 的硬件组成、工作原理、指令系统、编程规则、编程方法, 以及 CX-P 软件的应用技术等方面进行了完整和系统的描述。本节将介绍几个常用的应用实例。

### 6.3.1 自动定时搅拌系统

自动定时搅拌系统如图 6.46 所示, 该搅拌系统的动作过程如下:

初始状态是出料阀门 A 关闭, 然后进料阀门 B 打开, 开始进料, 液面开始上升。当液面上升到传感器 S1 时, 其常开触点接通时, 搅拌器开始搅拌。搅拌 5 min 后, 停止搅拌, 打开出料阀门 A。当液面下降到传感器 S2 时, 其常开触点闭合。关闭出料阀门 A, 又重新打开进料阀门 B, 开始进料, 重复上述过程。

该控制系统的 I/O 分配见表 6.10。功能图如图 6.47 所示, 梯形图如图 6.48 所示。



6.3.2 定位及减速控制

一台电动机拖动送料机构，通过脉冲检测装置，控制电动机的减速和停止，实现对原材料的剪切。对控制提出的要求如下：

- ① 用多齿凸轮与电动机联动，同时用接近开关(或光电开关)检测齿轮数，产生的脉冲输入至 PLC 的计数器。
  - ② 当计数器计到 4900 个脉冲时减速，到 5000 个脉冲时停机。
  - ③ 当电动机转动了 5000 个脉冲时，刀具下降将材料切断，同时将计数器复位。
- 系统采用 CP1H PLC 控制，输入/输出点的分配见表 6.11，程序如图6.49所示。

表 6.11 I/O 分配

输 入		输 出	
启动按钮	0.00	电动机运行	100.00
停止按钮	0.01	电动机减速	100.01
接近开关	0.02	电动机停止	100.02

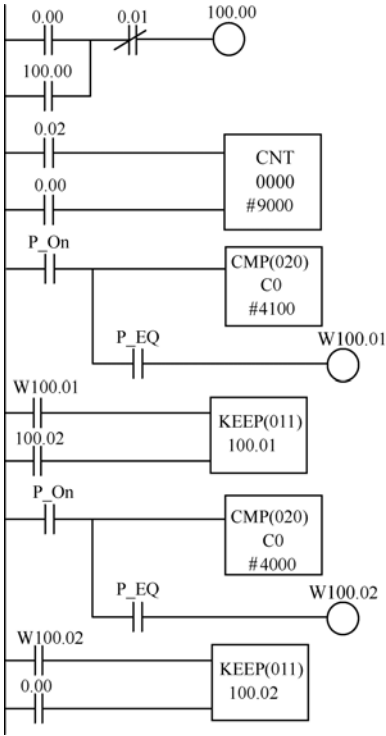


图 6.49 定位及减速控制程序

6.3.3 汽车自动清洗机

汽车自动清洗机的动作过程如下：

- ① 按下启动按钮时，打开喷淋器阀门，同时清洗机开始移动。
- ② 当检测到汽车到达刷洗距离时，启动旋转刷子开始刷洗汽车。
- ③ 当检测到汽车离开清洗机时，停止清洗机移动，停止刷子旋转和关闭阀门。
- ④ 当按下停止开关时，任何时候都可以停止所有的动作。

对整个控制系统的设计如下：

该控制系统的 I/O 分配见表 6.12。功能图如图6.50所示，程序如图6.51所示。

表 6.12 I/O 分配

输 入		输 出		步	
启动按钮	0.00	清洗机运动电动机	100.00	0 步	W0.00
汽车检测开关	0.01	喷淋器阀门	100.01	1 步	W0.01
停止按钮	0.02	刷子电动机	100.02	2 步	W0.02
—	—	—	—	3 步	W0.03

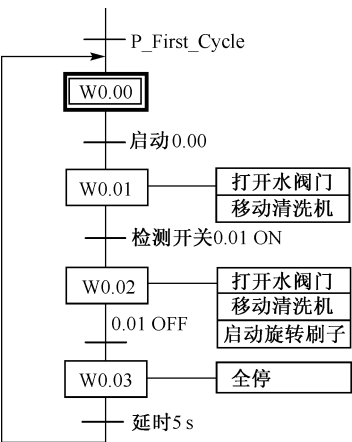


图 6.50 汽车自动清洗顺序功能图

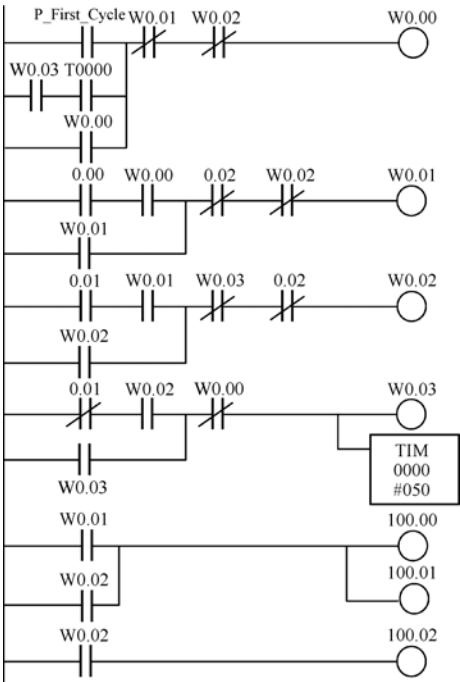


图 6.51 汽车自动清洗程序

6.3.4 霓虹灯控制电路

设计一个霓虹灯控制电路，该电路可以控制霓虹灯 L1、L2 和 L3 按照如下方式亮灭。

- ① 灯 L1 亮 2 s；
- ② 灯 L2 亮 2 s；
- ③ 灯 L3 亮 2 s；
- ④ 三个灯都灭 2 s；
- ⑤ 三个灯都亮 2 s；
- ⑥ 三个灯都灭 2 s。

该控制系统的 I/O 分配见表 6.13。功能图如图6.52所示，程序如图6.53所示。

表 6.13 I/O 分配

内 部 输 入		输 出		步	
启动	P_First_Cycle	灯 L1	100.00	第 0 步	W0.00
—	—	灯 L2	100.01	第 1 步	W0.01
—	—	灯 L3	100.02	第 2 步	W0.02
—	—	—	—	第 3 步	W0.03
—	—	—	—	第 4 步	W0.04
—	—	—	—	第 5 步	W0.05

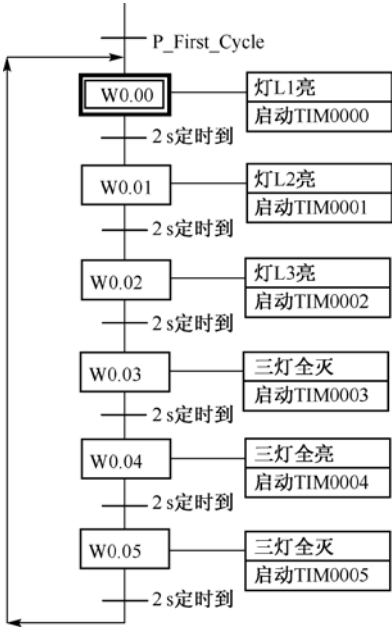


图 6.52 顺序功能图

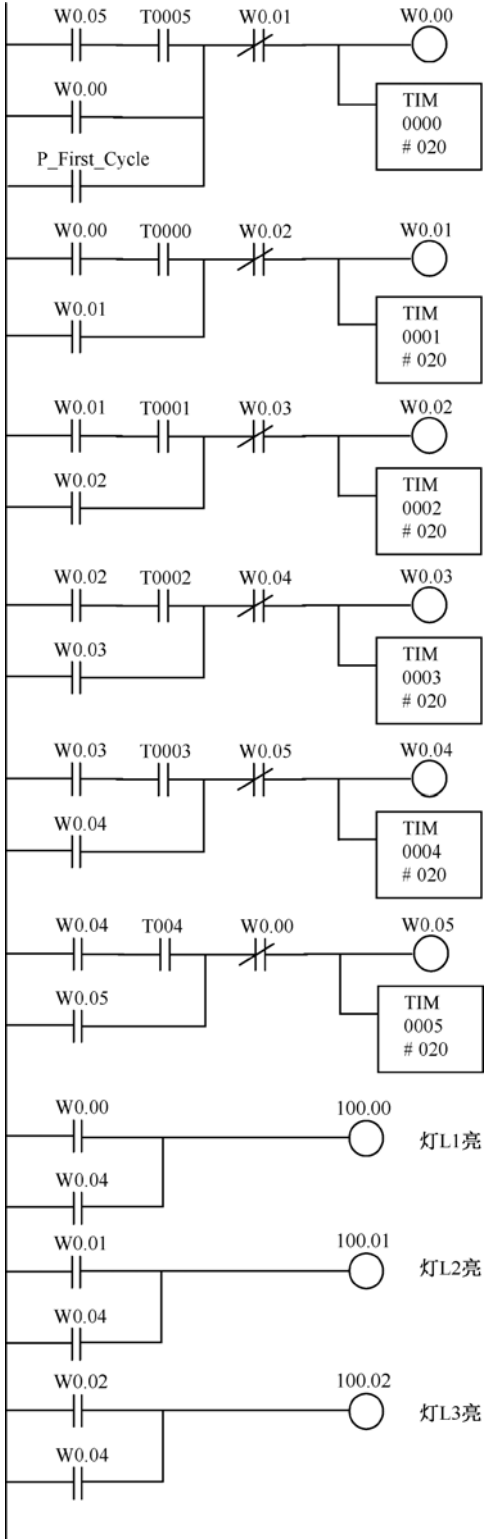


图 6.53 霓红灯控制程序

6.3.5 三层载货电梯控制

某三层大楼安装载货电梯一部，楼层的每一层安装呼叫按钮一个，呼叫灯一个。电梯的升降由一台电动机拖动，电动机正转电梯上升，反之，电梯下降；每层设置有行程开关，当电梯到达时，行程开关触点接通，电梯的动作如下：

- ① 电梯在一层或二层时，三层呼叫，电梯上升到三层。

② 电梯在二层或三层时，一层呼叫，电梯下降到一层。

③ 电梯在一层时，二层呼叫，电梯上升到二层。

④ 电梯在三层时，二层呼叫，电梯下降到二层。

⑤ 电梯在一层时，二层呼叫后，三层又呼叫，电梯上升到二层，停 2 s 后，继续上升到三层。

⑥ 电梯在三层时，二层呼叫后，一层又呼叫，电梯下降到二层，停 2 s 后，继续下降到一层。

⑦ 电梯在一层时，三层呼叫后，电梯到达二层前，二层呼叫， 电梯在二层停 2 s 后，继续上升到三层。若是到达二层以后，则不理睬二层呼叫。

⑧ 电梯在三层时，一层呼叫后，电梯到达二层前，二层呼叫， 电梯在二层停 2 s 后，继续下降到一层。若是到达二层以后，则不理睬二层呼叫。

⑨ 电梯在二层时，一层呼叫后，三层也呼叫，则电梯下降到一层，在一层停 2 s 后再上升到三层。

⑩ 电梯在二层时，三层呼叫后，一层也呼叫，则电梯上升到三层，在三层停 2 s 后再下降到一层。

⑪ 电梯在上升或下降途中，任何反方向的呼叫无效。
- 该控制系统的设计如下。
- ① I/O 分配见表 6.14。

表 6.14 安装载货电梯的 I/O 分配

输 入		输 出		步	
一层呼叫	0.01	电梯下降	100.00	程序 a 的第一步	W3.01
二层呼叫	0.02	电梯上升	100.01	程序 b 的第一步	W1.01
三层呼叫	0.03	一层指示灯	101.01	程序 c 的第一步	W3.11
一层行程开关	1.01	二层指示灯	101.02	程序 d 的第一步	W1.11
二层行程开关	1.02	三层指示灯	101.03	程序 e 的第一步	W2.01
三层行程开关	1.03	—	—	程序 f 的第一步	W2.11

- ② 载货电梯的功能图如图6.54 所示。

图6.54(a)是电梯在三层时，一层呼叫和一层呼叫后二层又呼叫的情况。

图6.54(b)是电梯在一层时，三层呼叫和三层呼叫后二层又呼叫的情况。

图6.54(c)是电梯在三层时，二层呼叫和二层呼叫后一层又呼叫的情况。

图6.54(d)是电梯在一层时，二层呼叫和二层呼叫后三层又呼叫的情况。

图6.54(e)是电梯在二层时，一层呼叫后三层又呼叫的情况。

图6.54(f)是电梯在二层时，三层呼叫后一层又呼叫的情况。
- ③ 图 6.54 所示功能图相对应的梯形图如图 6.55 所示。图 6.55(g) 是该电梯控制系统的输出电路。

为了实验方便，增加了楼层到位指示灯的梯形图。

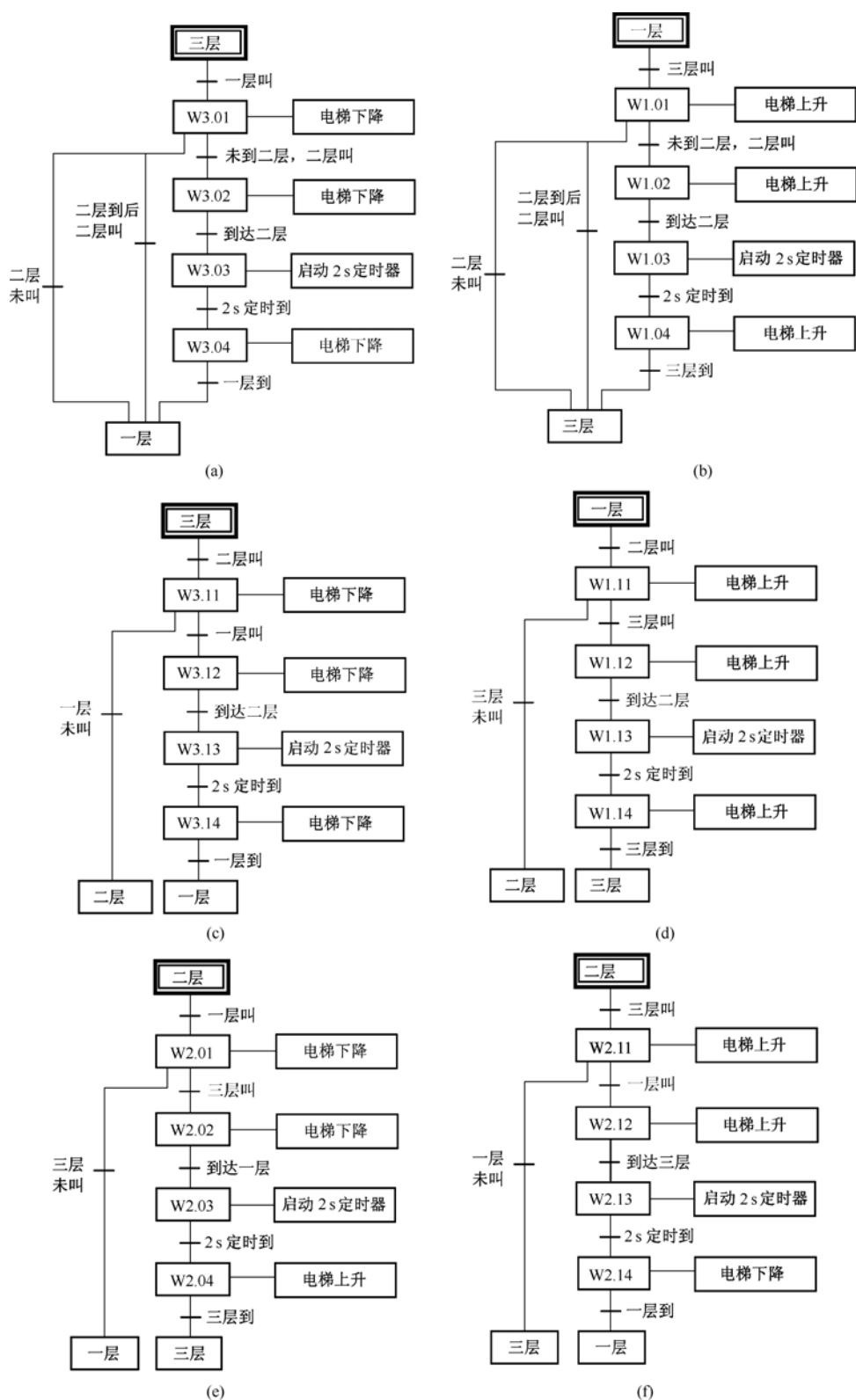


图 6.54 顺序功能图

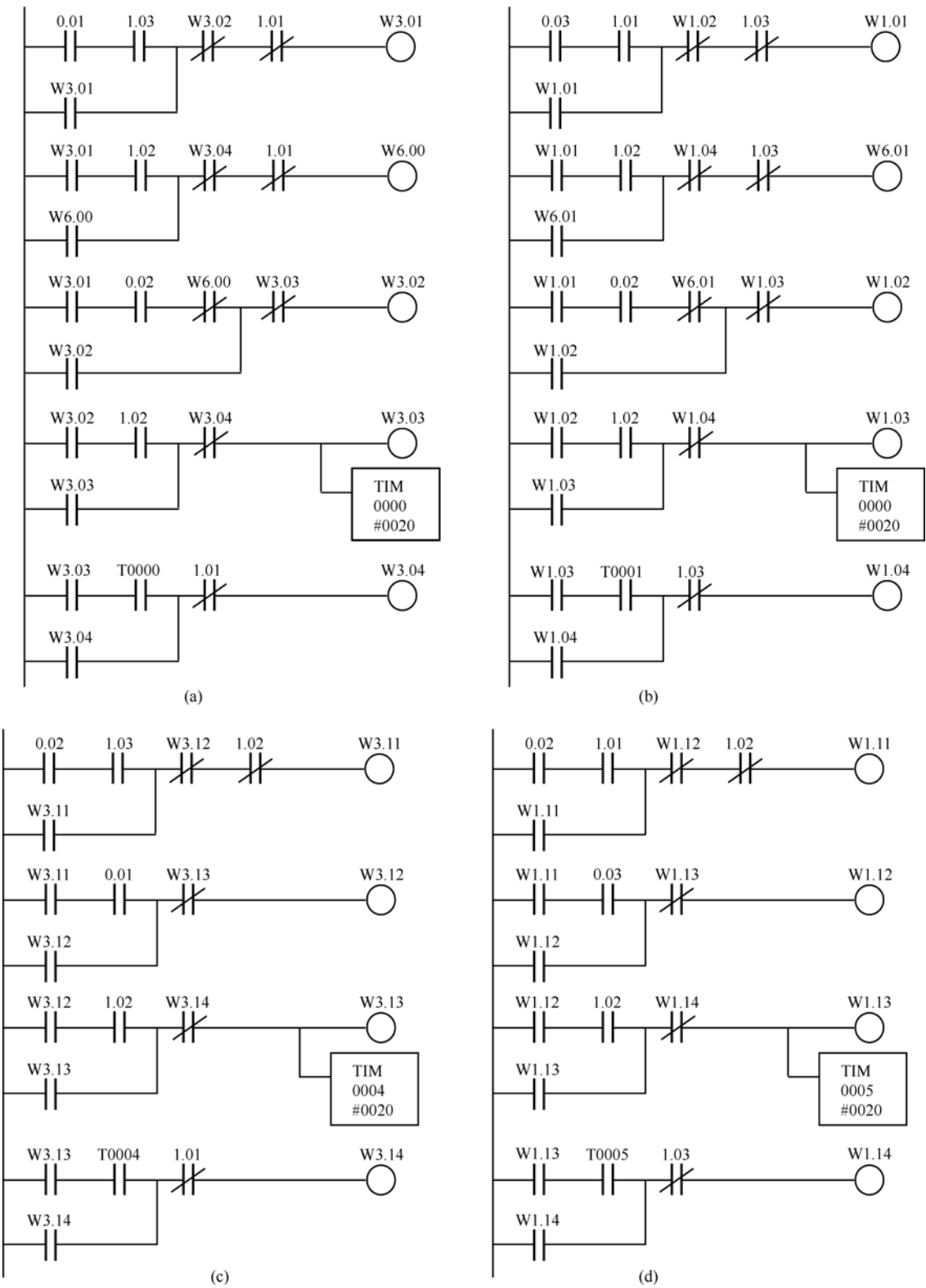


图 6.55 梯形图



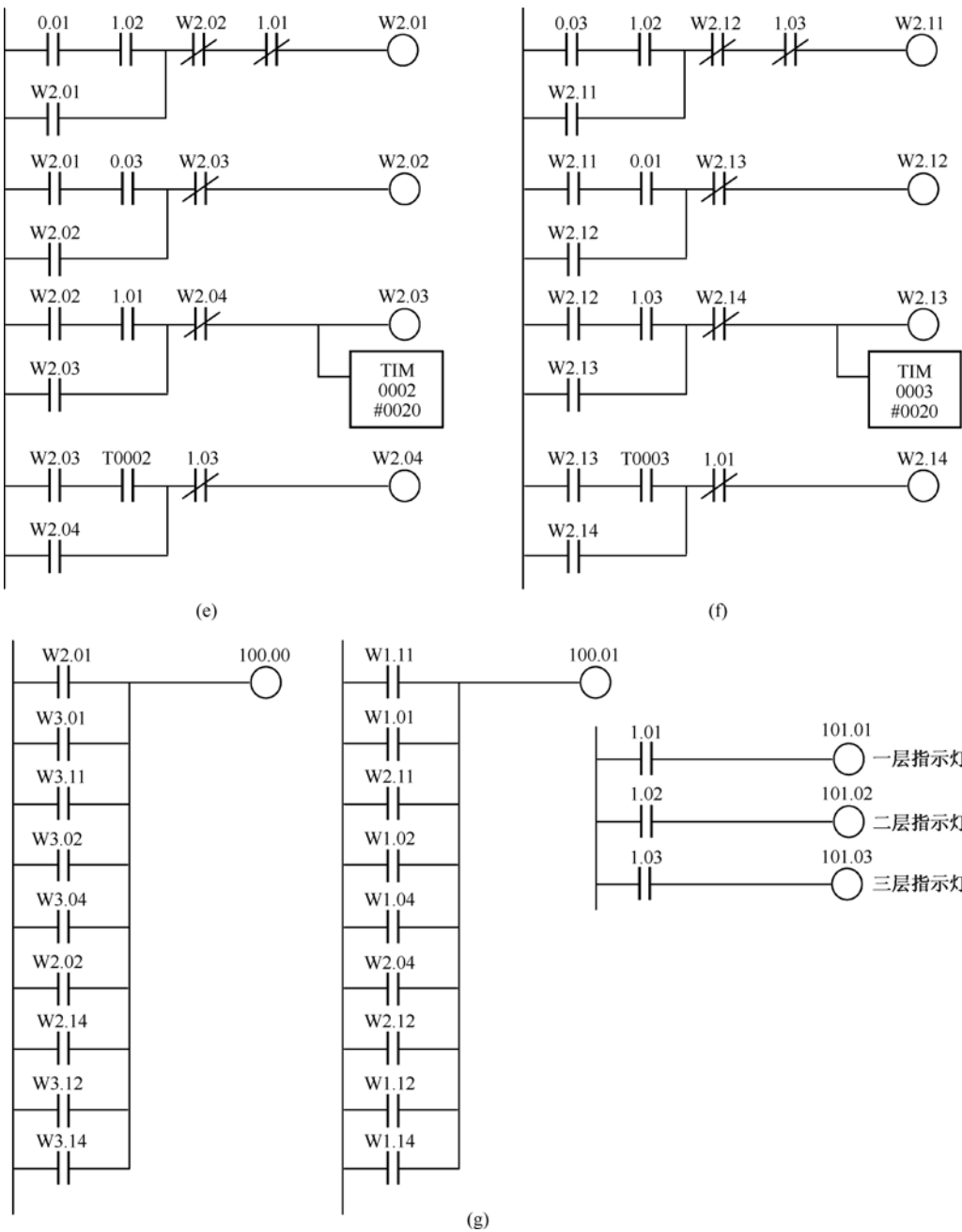


图 6.55 梯形图(续)

6.3.6 自动售货机的控制

自动售货机的动作如下。

- ① 此自动售货机可以投入 1 元、5 元和 10 元硬币。
- ② 自动售货机可售两种饮料，果汁每瓶 12 元，啤酒每瓶 15 元。
- ③ 当投入的硬币总值等于或超过 12 元时，果汁指示灯亮；当投入的硬币总值等于或超过 15 元时，果汁和啤酒指示灯都亮。

- ④ 当果汁指示灯亮时，按下果汁按钮，则售货机输出果汁。
- ⑤ 当啤酒指示灯亮时，按下啤酒按钮，则售货机输出啤酒。
- ⑥ 若投入的硬币总值超过所选饮料的价值时，售货机计算出余额，并且以币值为 1 元的硬币退还给消费者。

根据上述售货机的动作，可以想象售货机应该有投入硬币值计算，确认可以购买的饮料种类，根据选择输出饮料，计算余额，根据余额输出硬币给消费者等步骤。

根据动作要求，自动售货机的功能图如图6.56所示。自动售货机的 I/O 分配见表 6.15。

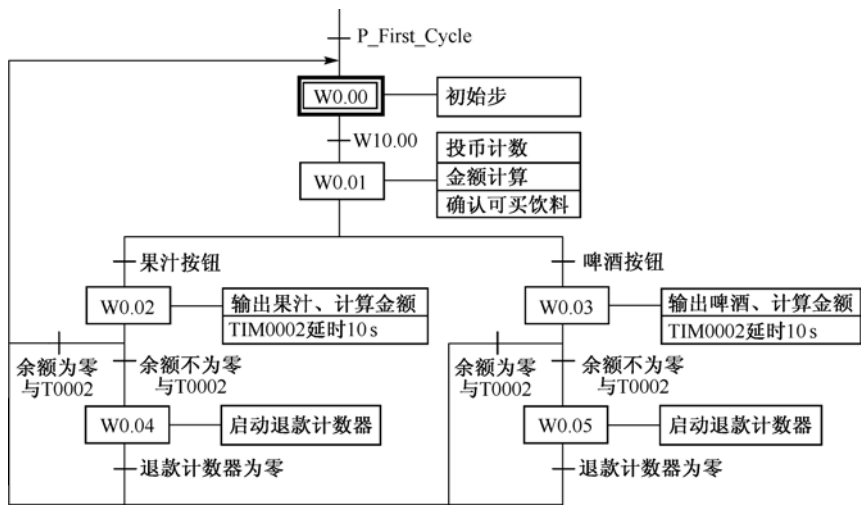


图 6.56 自动售货机的功能图

表 6.15 自动售货机的 I/O 分配

输 入		输 出	
1 元投币计数	0.01	大于 12 元指示灯	100.00
5 元投币计数	0.02	大于 15 元指示灯	100.01
10 元投币计数	0.03	输出果汁	100.02
果汁选择按钮	0.05	输出啤酒	100.03
啤酒选择按钮	0.06	—	—
找钱 1 元硬币计数	0.07	—	—
启动按钮	1.00	—	—
停止按钮	1.01	—	—

梯形图中使用的 DM 数据区见表 6.16。

表 6.16 DM 数据区分配

内容	5 元硬币 总钱数	10 元硬币 总钱数	5 元和 10 元 硬币总钱数	1 元、5 元和 10 元 硬币总钱数	余额
数据区	D5	D10	D15	D16	D20

自动售货机的梯形图如图6.57所示。

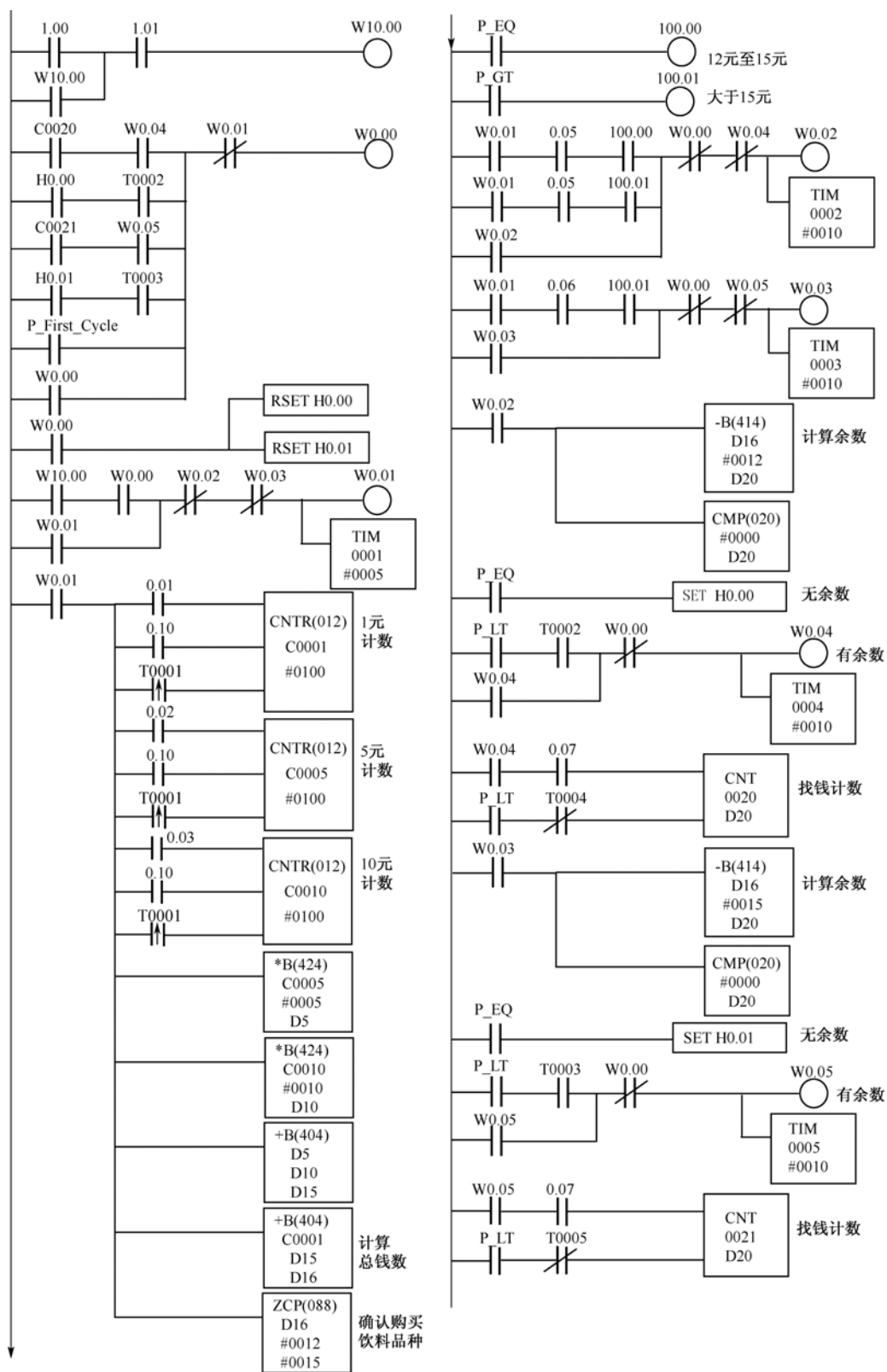


图 6.57 自动售货机梯形图

6.3.7 送料小车的控制

送料小车工作示意图如图6.58所示，初始状态小车空车停在行程开关 ST1 处，按一下启动按钮，小车在 ST1 处装料；15 s 后装料结束，开始右行；碰到行程开关 ST2 后停下来卸料；10 s 后左行，碰到行程开关 ST1 又停下来装料；15 s 后装料结束，开始右行，碰到行程开关 ST3 后停下来卸料，10 s 后左行，碰到行程开关 ST1 又停下来装料。这样循环工作，直到按一下停止按钮，小车在完成最后一个周期的工作后空车停在 ST1 处。

① 系统的 I/O 分配见表 6.17。

图 6.58 送料小车工作示意图

输 入		输 出		步 (0 步等待步)	
ST1	0.01	右行接触器	100.00	第 0 步	W0.00
ST2	0.02	左行接触器	100.01	第 1 步	W0.01
ST3	0.03	装料电磁阀	100.02	第 2 步	W0.02
启动按钮	0.04	卸料电磁阀	100.03	第 3 步	W0.03
停止按钮	0.05	—	—	第 4 步	W0.04
—	—	—	—	第 5 步	W0.05
—	—	—	—	第 6 步	W0.06
—	—	—	—	第 7 步	W0.07
—	—	—	—	第 8 步	W0.08

② 功能图如图6.59所示。

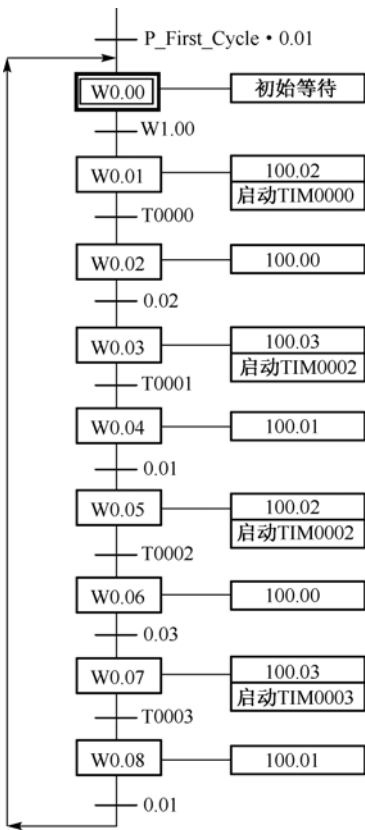


图 6.59 功能图

注意该系统是一种连续工作方式，即在初始点 ST1 处，按一下启动按钮，系统进行连续循环工作，直到按一下停止按钮，系统在完成最后一个周期的工作后，停止在 ST1 处。0 步(W0.00 步)为等待步，W1.00 是连续工作状态位，W0.00 到 W0.01 的转换条件为 W1.00=1。

### ③ 设计梯形图。

根据功能图画出梯形图，如图6.60所示。

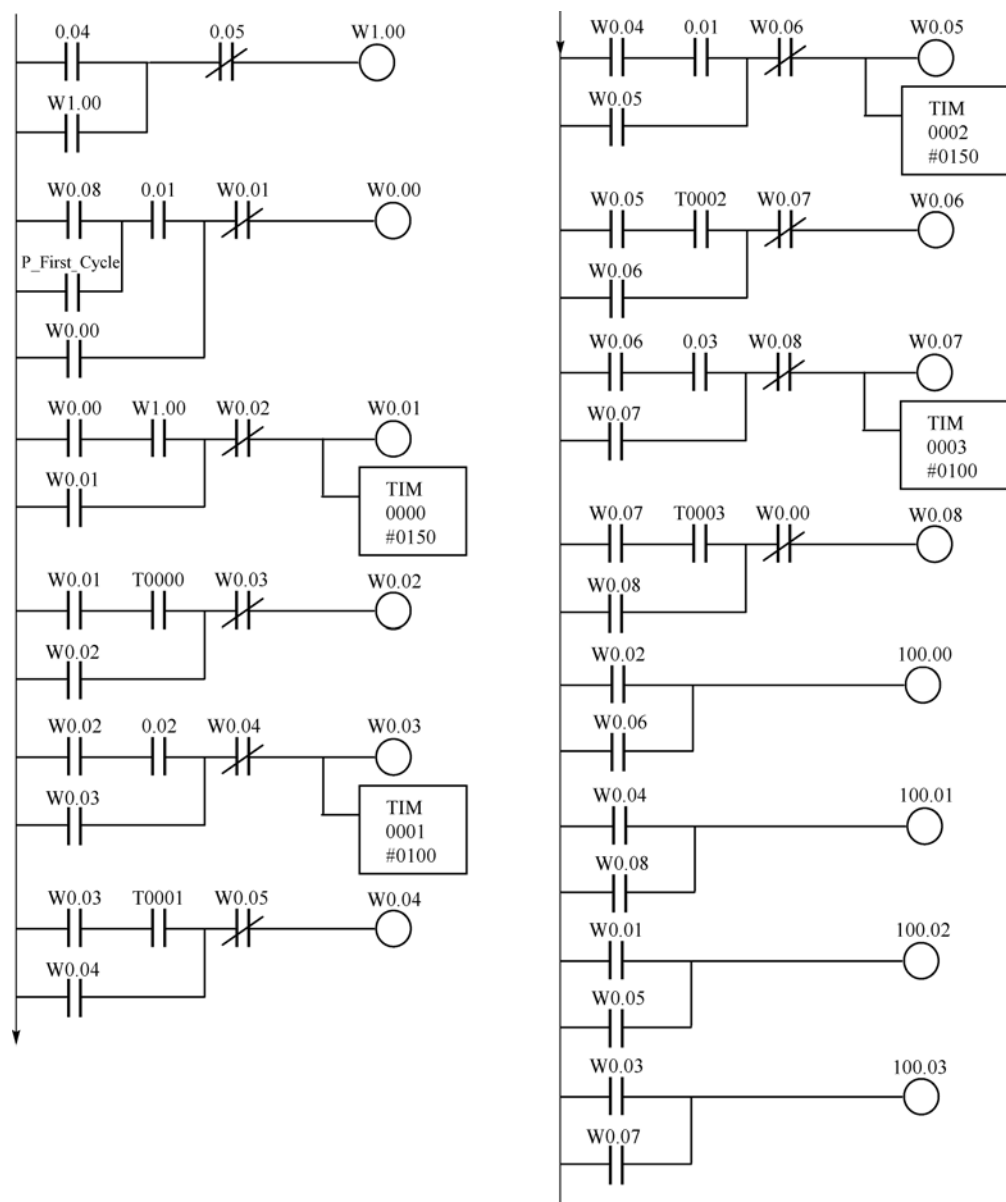


图 6.60 梯形图

## 6.3.8 传送带控制系统

用传送带传送长物体的控制系统如图6.61所示。为了减少传送带的运行时间，采用分段传送方式，A、B 为两条传送带，GK1、GK2 为两个光电开关，工作过程如下：



图 6.61 传送带示意图

不运行；当长物体前端靠近 GK1 时，A、B 都运行；当长物体后端离开 GK1 时，B 运行，A 不运行；当长物体后端离开 GK2 时，A、B 都不运行。

- ① I/O 分配见表 6.18。
- ② 功能图如图6.62 所示。 $\overline{0.00}$  表示转换条件为 0.00 OFF，0.01 ↓表示转换条件为 0.01 的下降沿微分。
- ③ 设计梯形图，梯形图如图6.63 所示。

表 6.18 I/O 分配表

输 入		输 出	
GK1	0.00	A 线圈	100.00
GK2	0.01	B 线圈	100.01
启动按钮	0.02	—	—
停止按钮	0.03	—	—

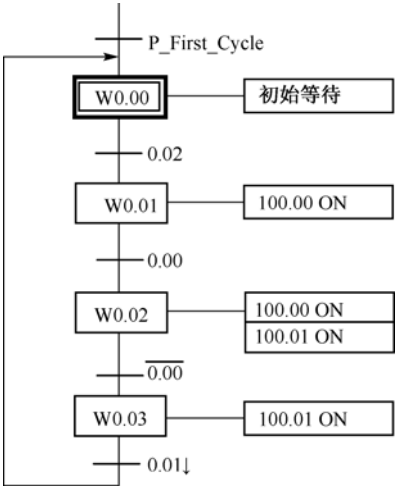


图 6.62 功能图

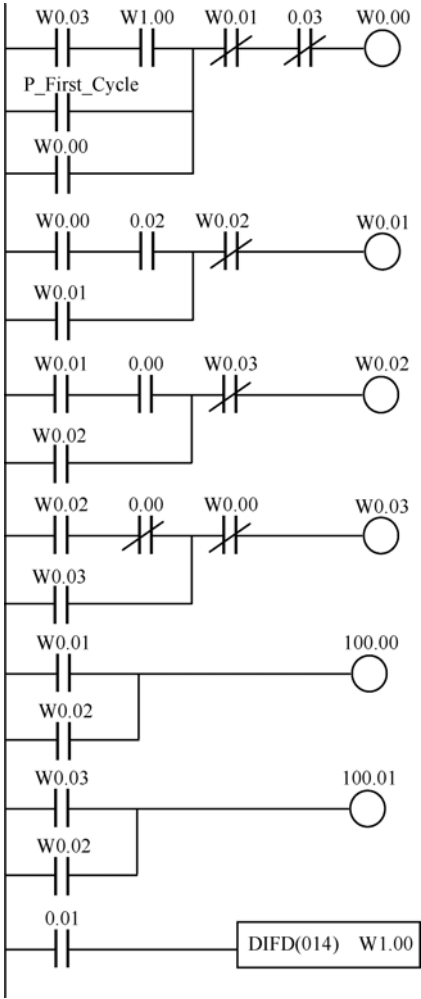


图 6.63 梯形图

6.3.9 配料槽控制系统

某产品由 3 种原料按一定比例混合配制，3 种原料分别放在 A、B 和 C 3 个槽中。配料槽工作示意图如图6.64 所示。

- ① 按下启动按钮后，3 个槽阀门一起打开，3 种原料流入主槽，但各槽阀门粗细不一，流入速度也不相同。每个槽都有设定容量的原料。
- A 槽：每隔 0.02 s，流入一个单位质量的原料。

B 槽：每隔 0.01 s，流入一个单位质量的原料。  
C 槽：每隔 0.2 s，流入一个单位质量的原料。

② 主槽容量显示存于 H0 中，以备显示。

③ 各槽排空后，各自料空指示灯亮，3 个灯都亮后，主槽灯亮表示料满。

④ 3 槽全部排出 5 s 后，4 个灯全灭，进入下一道工序。

I/O 分配见表 6.19，存储区见表 6.20。

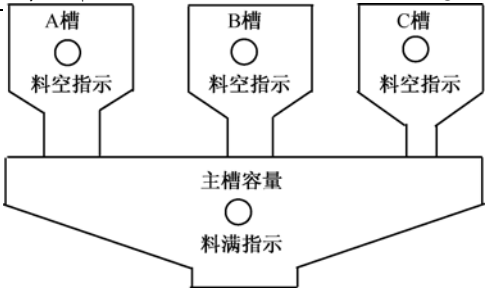


图 6.64 配料槽工作示意图

表 6.19 I/O 分配表

输 入		输 出	
启动按钮	0.00	A 阀	100.00
—	—	B 阀	100.01
—	—	C 阀	100.02
—	—	A 槽料空指示灯	100.03
—	—	B 槽料空指示灯	100.04
—	—	C 槽料空指示灯	100.05
—	—	主槽料满指示灯	100.06

表 6.20 程序中用到的存储区

内 容	A 槽原料量	B 槽原料量	C 槽原料量	主 槽 容 量
存储区	D0	D1	D2	H0

配料槽梯形图如图 6.65 所示。

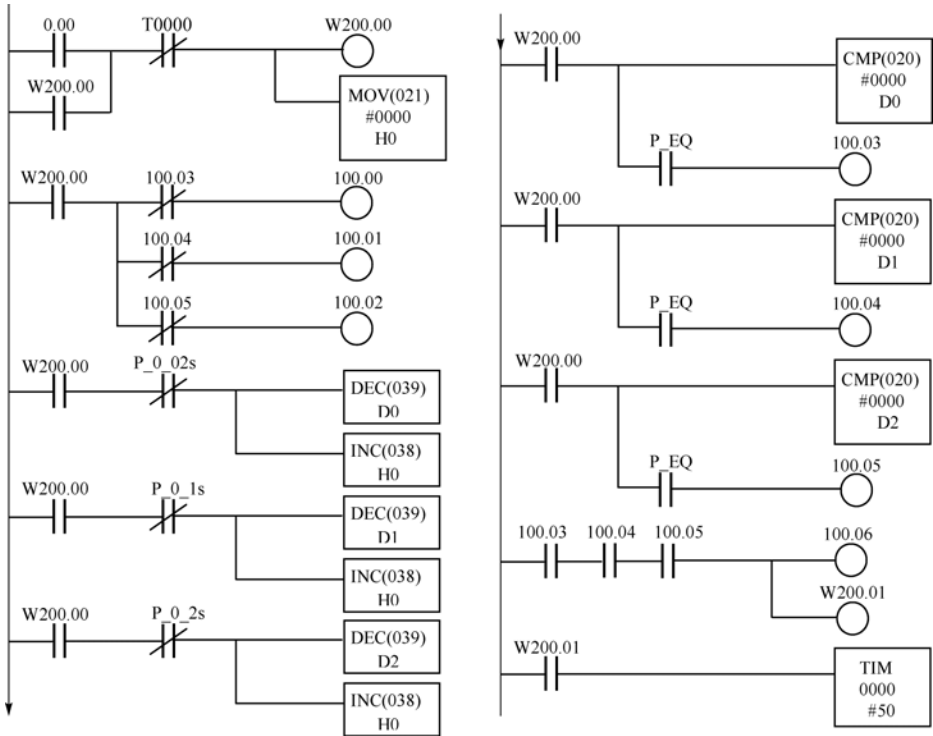


图 6.65 配料槽梯形图

6.3.10 U 形板折板机的 PLC 控制

U 形板折板机可以把金属板料折成如图6.66(a)所示的形状。U 形板折板机的基本组成结构如图 6.66(b)所示。模板装在模板座上，模板两端的形状不同，加工出来的 U 形板料的折角形状就不同，折角的大小由左右的限位开关的位置决定。根据不同要求，模板可以随时更换。在气压的推动下，模板与模板座一起下移或上移。被加工的金属板料放在平台上，当模板下移压紧板料后，工作平台上的左右折板在气压机构的推动下，可向上折，也可以折回，把板料加工成 U 形。

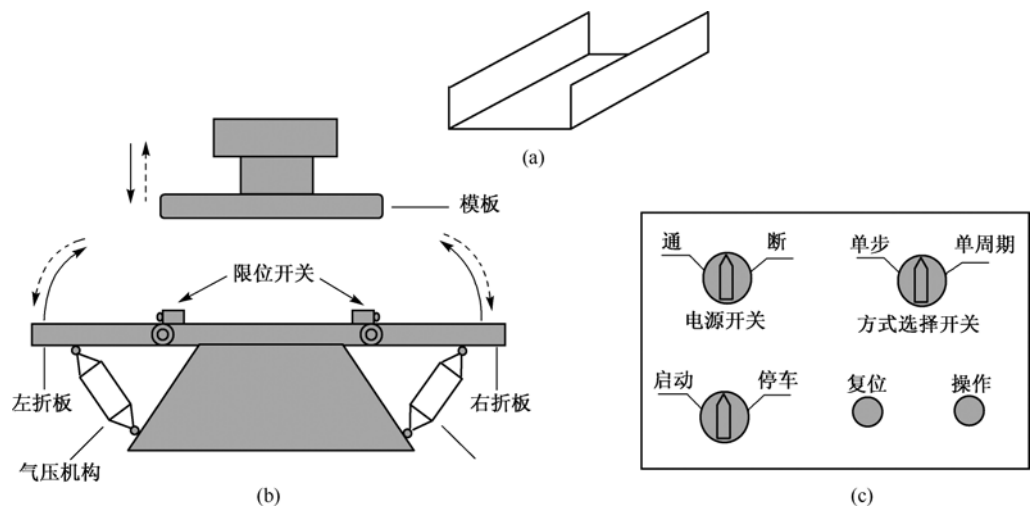


图 6.66 U 形板折板机的结构及操作盘示意图

1. 加工工艺过程

U 形板折板机的加工工艺过程如下：

- ① 当模板上移到位，左右折板返回原位时，将裁好的金属板料放在工作平台上。
- ② 启动设备，模板开始下移，下移到位时压紧板料。
- ③ 接着左右折板上折，上折到位压动左右限位开关时，停止上折并保持压力 2 s。
- ④ 接着左右折板返回，折板返回原位时自停。
- ⑤ 接着模板上移，上移到原位自停。
- ⑥ 取下 U 形板，一块板料的加工过程结束。

2. 控制要求

- ① 折板机的控制分为单步和单周期两种方式。单步方式时，按一次操作按钮执行一个工作步。单周期方式时，按一次操作按钮连续完成上述加工过程后自停，加工过程不循环。
- ② 模板下移和上移是用双线圈的电磁阀控制的。当一个线圈通电时模板上移，当另一个线圈通电时模板下移。
- ③ 左右折板上折和折回各用一个两线圈的电磁阀控制，当电磁阀的一个线圈通电时，折板上折，当另一个线圈通电时折板返回。
- ④ 两个折板必须都上折到位才能开始保持压力。
- ⑤ 折板机运行过程中可以停机。完成一个加工过程自动停机时，模板应在上方返回原位，左右折板应返回到水平原位。



⑥ 停机再开机时，如果模板和左右折板都在原位，则可以开始进行板料加工。如果不在原位，要将它们复位到原位才可以开始加工。

⑦ 手动复位时，用复位按钮来控制模板和两个折板的复位，但模板上移和折板的折回动作不能同时启动，以免两者发生摩擦而损坏模板和折板。

⑧ 自动运行和手动操作过程中，若误按操作按钮时不应出现误动作。

3. 操作盘面板布置及 I/O 分配

(1) 操作盘的布置及各元件的作用

操作盘的布置如图 6.66(c) 所示。根据上述控制要求，操作盘上要设置：用于接通、断开 PLC 电源的开关(不占输入点)；工作方式选择开关(占 1 个输入点)，分单周期位和单步位；启动/停车开关(占 1 个输入点)；1 个复位按钮，若停机时模板和左/右折板不在原位，再开机时要进行复位；1 个操作按钮，手动和自动操作时，都要使用操作按钮。

(2) I/O 分配

本机用 40 点的 CP1H 控制。除了操作盘上介绍的输入元件外，左右折板的上折限位各用一个限位开关，模板下移的控制需要一个限位开关。模板的上升、下移及折板的上折、折回所需的电磁阀(均为双线圈)都是输出执行元件。由上述设置，PLC 的 I/O 分配见表 6.21。

表 6.21 I/O 分配

输 入		输 出	
操作按钮	0.00	模板下移电磁阀线圈	100.00
方式选择开关	0.01	左折板上折电磁阀线圈	100.01
复位按钮	0.02	右折板上折电磁阀线圈	100.02
启动/停车开关	0.03	左折板返回电磁阀线圈	100.03
左折板上折限位开关	0.04	右折板返回电磁阀线圈	100.04
右折板上折限位开关	0.05	模板上移电磁阀线圈	100.05
模板下移限位开关	0.06	—	—

4. 梯形图程序及其控制功能

折板机的 PLC 控制梯形图如图 6.67 所示。下面介绍各种操作方式时的控制功能以及误操作禁止的原理。

(1) 复位操作

开机时若模板和折板不在原位，要进行复位操作。将方式选择开关拨在单步位，0.01 为 OFF、自动运行被禁止。按住复位按钮 0.02 不放，其复位过程如下。

- ① 移位寄存器 W211 通道被复位。
- ② W200.04 被置位，TIM0002 开始计时，0.5 s 后将 H0.00 置位，为板料加工做好准备。
- ③ 100.03 和 100.04 均为 ON，两个折板开始返回。为了避免模板和折板相互摩擦，要自折板开始返回 1 s 后，再让模板启动上移，因此设置 TIM0004 定时 1 s。
- ④ TIM0004 定时 1 s 到，模板开始上移。
- ⑤ 当模板和折板都返回原位(目测)时，松开复位按钮。

(2) 单周期方式的加工过程控制

模板和折板都返回原位时，将剪好的板料放在平台上。

- ① 将方式开关拨在单周期位、常开触点 0.01 闭合，使 SFT 的移位脉冲输入端接通，而单步运行方式被禁止。

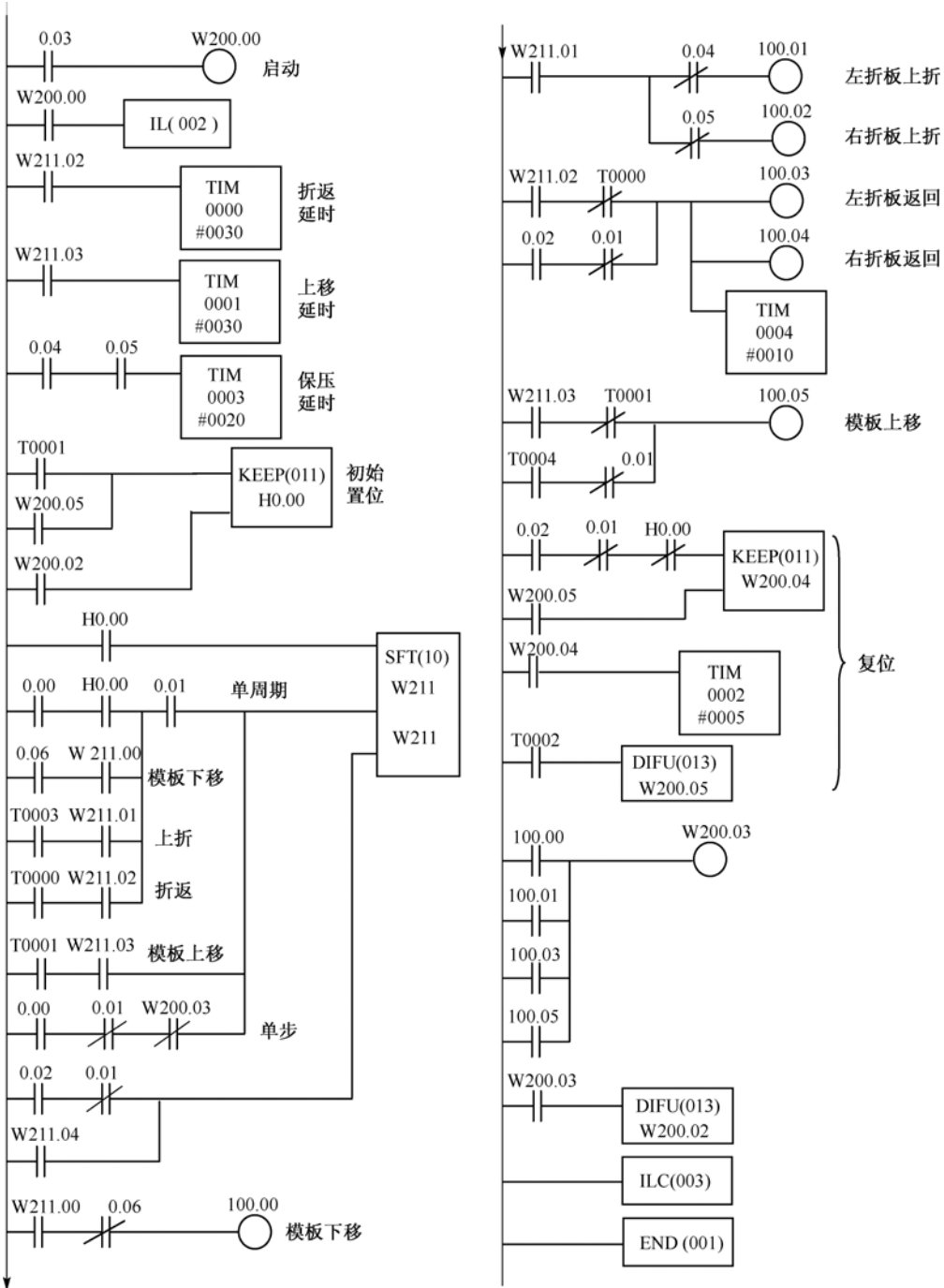


图 6.67 U 形板折板机的 PLC 控制梯形图

- ② 按一下操作按钮 0.00，第一次执行移位，W211.00、100.00 为 ON，则模板开始下移；W200.03 ON 并通过 W200.02 将 H0.00 复位，使下一次移位时移位输入是“0”。
- ③ 模板下移到位压下限位开关 0.06。其一，使 100.00 为 OFF，模板下移停止并压紧板料；其二，输入一个移位脉冲，使 W211.00 为 OFF，W211.01 为 ON，于是 100.01 和 100.02 为 ON，两折板开始上折。

④ 两折板上折到位, 压动限位开关 0.04 和 0.05, 其一, 使 100.01 和 100.02 均为 OFF, 折板上折停止; 其二, 当 0.04 和 0.05 均为 ON 时 TIM0003 开始计时, 保持压力 2 s。

⑤ TIM0003 计时到, 输入一个移位脉冲, 使 W211.01 为 OFF, W211.02 为 ON, 则 100.03 和 100.04 均为 ON, 折板开始返回, TIM0000 进行折板返回计时。

⑥ TIM0000 定时时间到, 输入一个移位脉冲, 使 W211.02 为 OFF, W211.03 为 ON, 100.03 和 100.04 均为 OFF, 折板返回停止; 100.05 为 ON 使模板开始上移, TIM0001 开始模板上移计时。

⑦ TIM0001 计时到, 输入一个移位脉冲, 使 W211.03 为 OFF, W211.04 为 ON。其一, 100.05 为 OFF, 模板上移停止; 其二, TIM0001 为 ON, 将 H0.00 置位, 为下一次加工做好准备; 其三, W211.04 为 ON 使移位寄存器复位。

到此为止, 加工一块板料的过程结束。下一块板料的加工与上述过程相同。

### (3) 单步操作

① 将方式开关拨在单步位, 常开触点 0.01 断开, 自动方式被禁止, 单步方式被允许。

② 按一下操作按钮, 0.00 为 ON 一次、输入一个移位脉冲, W211.00 和 100.00 均为 ON。其一, 模板开始下移; 其二, W200.03 ON 并通过 W200.02 将 H0.00 复位, 使下一次移位时输入的是“0”; 当模板下移到位压限位开关 0.06 时, 模板下移停止并压紧板料。

③ 再按一下操作按钮 0.00, 又输入一个移位脉冲, W211.00 为 OFF, W211.01 为 ON。使 100.01 和 100.02 均为 ON, 两折板开始上折。折板上折到位压限位开关 0.04 和 0.05, 则 100.01 和 100.02 均为 OFF, 使折板上折停止, 同时 TIM0003 开始计时保持压力 2 s。

④ 2 s 后再按一下操作按钮, 又输入一个移位脉冲, W211.01 为 OFF, W211.02 为 ON。于是折板开始返回, TIM0000 进行折板返回计时。TIM0000 计时到, 100.03 和 100.04 均为 OFF, 折板返回停止。

⑤ 再按一下操作按钮, 又输入一个移位脉冲, W211.02 为 OFF, W211.03 为 ON。于是模板开始上移, TIM0001 开始模板上移计时。TIM0001 计时到, 其一, 输入一个移位脉冲, 使 W211.03 为 OFF, W211.04 为 ON, 则 100.05 为 OFF, 模板上移停止; 其二, W211.04 为 ON 使移位寄存器复位; 其三, TIM0001 ON 将 H0.00 置位, 为下一次加工做好准备。

### (4) 误操作时误动作的禁止

① 单步操作时误动作的禁止。当按过一次操作按钮, 正在执行某步加工的过程中又误按了一下操作按钮时, 不会产生误动作。因为在每一步启动时按一次按钮 0.00 后, 对应各个步, 分别使 100.00、100.01、100.03、100.05 为 ON, 都能使 W200.03 为 ON, 其常闭触点断开。所以在某步运行过程中误按操作按钮 0.00 时, 不会再输入移位脉冲, 也就避免了误动作。

另外, 由手动开关产生的信号一般都可能产生抖动, 例如, 按一次按钮却连续发出几个脉冲信号。本例中, 为了避免由这种现象造成的误动作, 采取了如下措施: 将常闭触点 W200.03 与 0.00 串联, 由于 0.00 第一次为 ON 就使 W200.03 为 ON, 其常闭触点即断开, 这样 0.00 的后几次 ON 就不会起作用了。

② 单周期操作时误动作的禁止。当第一次按过操作按钮 0.00, 设备启动后, 100.00 为 ON, W200.03 为 ON, 并通过 W200.02 将 H0.00 复位。由于 0.00 与 H0.00 的常开触点串联后连接在移位寄存器的移位脉冲输入端, 在整个加工过程中, 由于 H0.00 的常开触点总是断开的, 即使误按操作按钮 0.00, 也不会产生错误的移位脉冲输入, 所以由误按操作按钮而产生的误动作不会出现。

在设备运行过程中, 其他运动部件误撞动操作按钮的情况也是难免的。由上述分析可知, 即使发生这种情况, 也能避免误动作。

③ 设备的零压保护。设备运行过程中若断电, 再复电时不应自行启动。本例, 在 PLC 外部设置了零压保护措施(这里没画出其外部电路)。所以, 若加工过程中断电, 再复电时设备是不会自行启动的。

6.3.11 并行序列顺序控制

某电液控制系统中有两个动力头，其工作流程图如图6.68所示。控制要求为：

- ① 系统启动后，两个动力头便同时开始按流程图中的工步顺序运行。从它们都退回原位开始延时 10 s 后，又同时开始进入下一个循环的运行。
- ② 若断开控制开关，各动力头必须将当前的运行过程结束，即退回原位后才能自动停止运行。
- ③ 各动力头的运动状态取决于电磁阀线圈的通、断电，它们的关系见表6.22和表6.23。表中的“+”表示该电磁阀的线圈通电，“-”表示该电磁阀的线圈不通电。

由工作流程图可知各动力头的工作步数和转换条件。每个动力头的步与步之间的转换是单向进行的，最后转换到同一个步上。由于两个动力头退回原位的时间存在差异，所以要设置原位等待步。这样，只有两个动力头都退回原位时定时器才开始计时，确保两个动力头同时进入下一个循环的运行。因此画两个动力头的控制过程功能图时，应是并行序列结构。

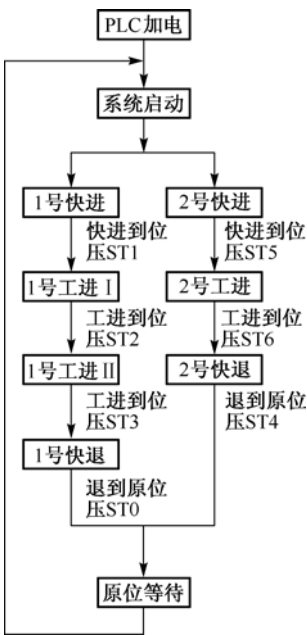


图 6.68 动力头工作流程图

表 6.22 1 号动力头

动作	YV1	YV2	YV3	YV4
快进	-	+	+	-
工进 I	+	+	-	-
工进 2	-	+	+	+
快退	+	-	+	-

表 6.23 2 号动力头

动作	YV5	YV6	YV7
快进	+	+	-
工进	+	-	+
快退	-	+	+

由工作流程图可以看出，本例需要一个启/停控制开关、7 个限位开关，它们是 PLC 的输入元件。由表 6.22 和表 6.23 可知，需要 7 个电磁阀，它们是 PLC 的输出执行元件。

如果选择机型为 CP1H 时，I/O 分配见表 6.24。

表 6.24 I/O 分配

输 入		输 出	
系统启动控制开关	0.00	电磁阀 YV1 线圈	100.01
1 号动力头原位限位 ST0	1.00	电磁阀 YV2 线圈	100.02
1 号动力头快进限位 ST1	1.01	电磁阀 YV3 线圈	100.03
1 号动力头工进 I 限位 ST2	1.02	电磁阀 YV4 线圈	100.04
1 号动力头工进 II 限位 ST3	1.03	电磁阀 YV5 线圈	100.05
2 号动力头原位限位 ST4	1.04	电磁阀 YV6 线圈	100.06
2 号动力头快进限位 ST5	1.05	电磁阀 YV7 线圈	100.07
2 号动力头工进限位 ST6	1.06	—	—

利用 W0 通道中的位做各工作步的控制位，画出功能图如图6.69所示。

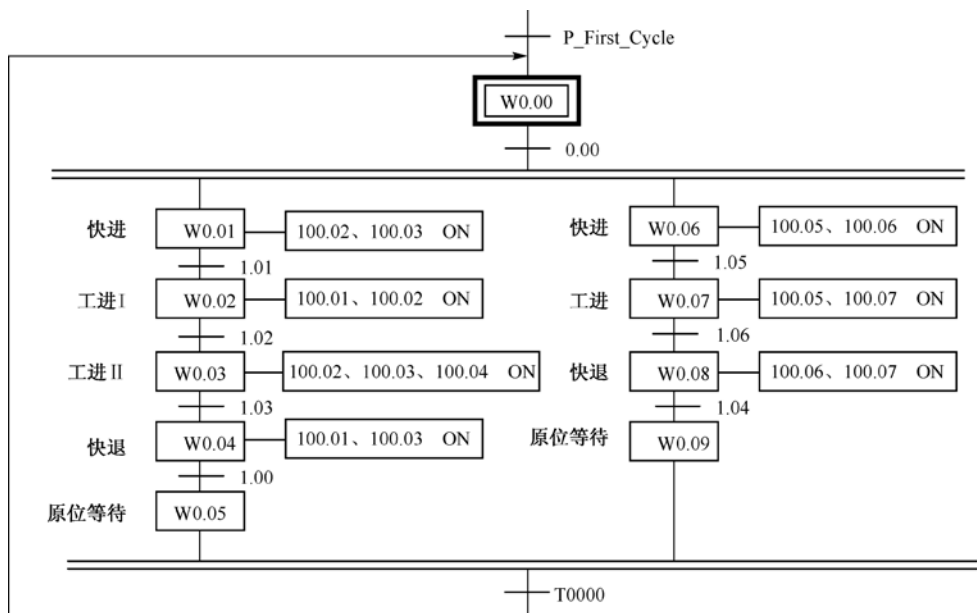


图 6.69 功能图

由功能图,按照前面介绍的方法很容易画出各步的梯形图。再根据各步应该接通的电磁阀线圈号,确定对应各步时电磁阀线圈的置位或复位状态,可画出如图6.70所示的梯形图。

下面简要介绍程序编写的思路。

① 在 PLC 加电后的第一个扫描周期  $P\_First\_Cycle$  为 ON,使初始步  $W0.00$  为 ON,为系统启动做好准备。

② 在一个循环的过程结束时,两个动力头一起在原位停留 10 s 后,步  $W0.00$  应能自动成为活动步,以使系统进入下一个循环的过程,所以将  $T0000$ (原位等待定时器)与  $P\_First\_Cycle$  并联。

③ 因为步  $W0.01$  和步  $W0.06$  是两个并行序列的首步,所以这两个步的活动条件都是  $W0.00$  和  $0.00$  的“与”。在一个循环的过程结束、 $W0.00$  成为活动步时,由于  $0.00$  始终为 ON,从而使步  $W0.01$  和步  $W0.06$  自动成为活动步,并开始重复前一个循环的过程。

④ 当两个动力头都回到原位、等待步  $W0.05$  和  $W0.09$  都成为活动步时,  $TIM000$  才开始计时。在定时时间到、步  $W0.00$  成为活动步时,等待步  $W0.05$  和  $W0.09$  才变为不活动步。

⑤ 对应每一个工作步,要对控制相关电磁阀的输出位进行置位或复位。例如,在  $W0.01$  成为活动步时,要将  $100.02$  和  $100.03$  置位(电磁阀  $YV2$ 、 $YV3$  线圈通电),使 1 号动力头快进。在等待步  $W0.05$  和  $W0.09$  为活动步时,将相关电磁阀线圈的输出位进行复位,以保证下一个循环时动力头不会发生错误的动作。例如,在  $W0.05$  成为活动步时,将  $100.01$  和  $100.03$  复位,使 1 号动力头进入等待状态,在  $W0.09$  成为活动步时,将  $100.06$  和  $100.07$  复位,使 2 号动力头进入等待状态。

顺序控制设计法有一定的规律可循,所编写的程序易读、易检查、易修改,是常用的设计方法之一。使用顺序控制设计法的关键有三条:一是理顺动作顺序,明确各步的转换条件;二是准确地画出功能图;三是根据功能图准确地画出相应的梯形图,最后再根据某些特殊功能要求,添加部分控制程序。要想用好顺序控制设计法,重要的是熟练掌握功能图的画法,以及根据功能图画出相应梯形图的方法。

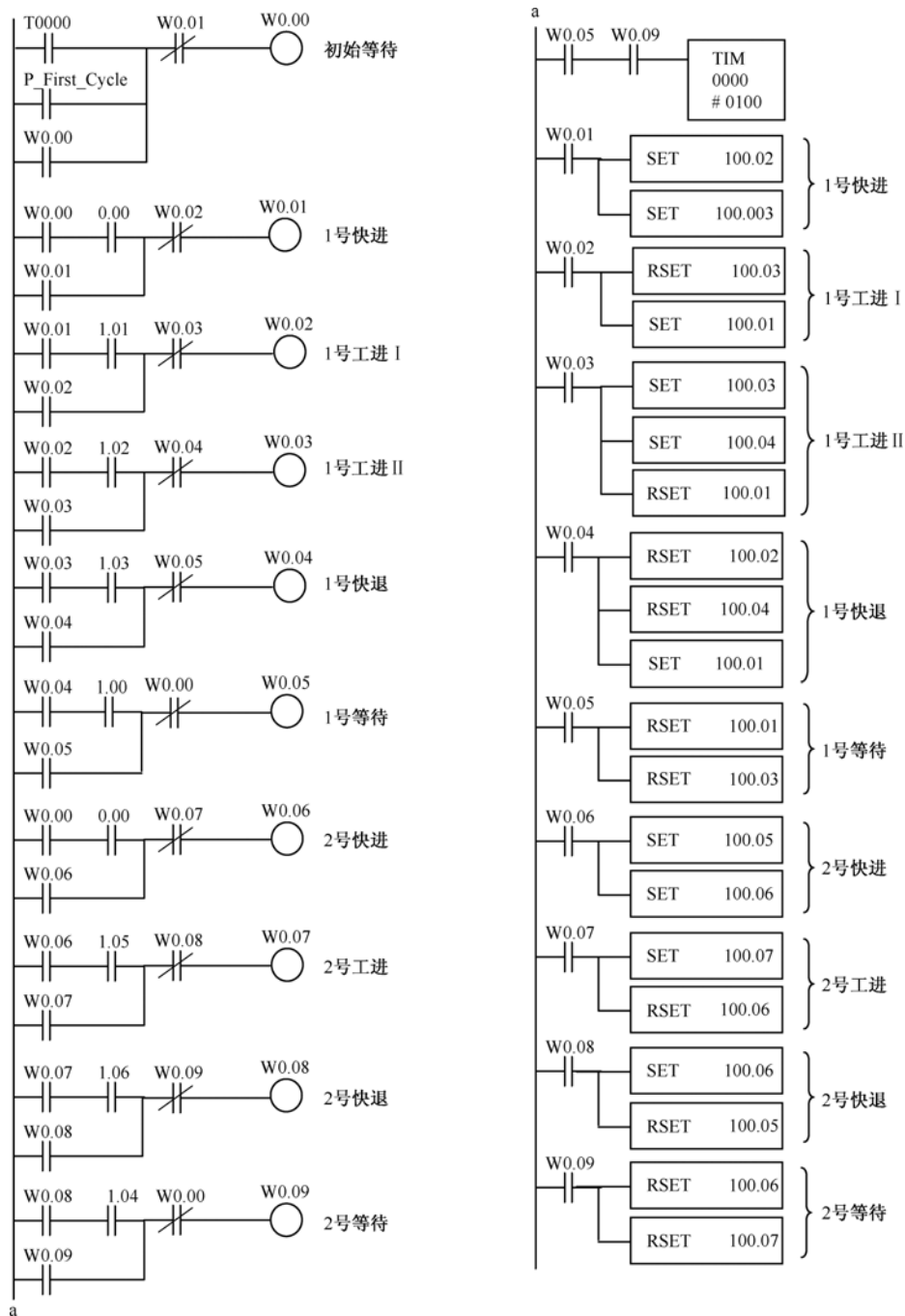


图 6.70 动力头控制梯形图

习 题

1. 某广告牌上有 5 个字，每个字显示 0.5 s 后，5 个字一起显示 1 s，然后全灭。0.5 s 后再从第一个字开始显示，重复上述过程。试用 CP1H PLC 实现上述控制功能，作 I/O 分配，并设计梯形图程序。
2. 3 台电动机 M1、M2、M3 按下面的顺序启动和停车：启动时，M1、M2 同时启动，4 分钟后 M3 才能启动；停车时，M3 必须先停，M3 停后 5 分钟 M1、M2 同时停。

要求：选用 CP1H PLC 实现上述电动机的控制，作 I/O 分配，画出 PLC 外部的接线图及电动机的主电路图，并设计梯形图程序。

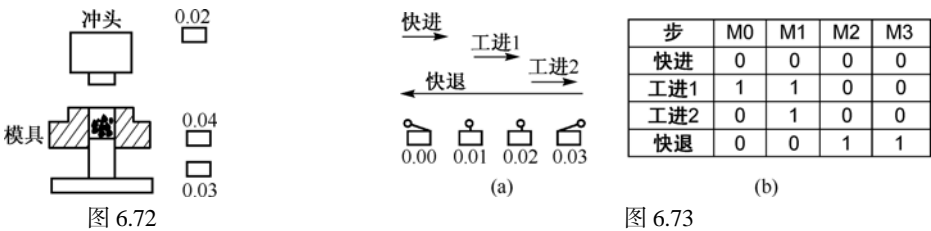
3. 4 台电动机的运行状态用 L1、L2、L3 三个指示灯显示。只有 1 台电动机运行时 L2 亮；2 台电动机运行时 L1 亮；3 台以上电动机运行时 L3 亮；都不运行时 3 个灯都不亮。

要求：选用 CP1H PLC 实现上述电动机的控制，作 I/O 分配，画出 PLC 外部的接线图及电动机的主电路图，并用逻辑设计法设计梯形图程序。

4. 4 个电磁阀的线圈 A、B、C、D 按图 6.71 所示的时序循环通断电(高电平为通)。请用时序图和逻辑设计法，设计出实现该控制的梯形图程序。

5. 粉末冶金制品压制机如图 6.72 所示，装好粉末后，按下启动按钮 0.00，冲头下行。将粉末压紧后，压力继电器 0.01 接通。保压延时 5 s 后，冲头上行至 0.02 接通。然后模具下行至 0.03 接通。取走成品后，工人按按钮 0.05，模具上行至 0.04 接通，系统返回初始状态。画出顺序功能图，设计出梯形图。

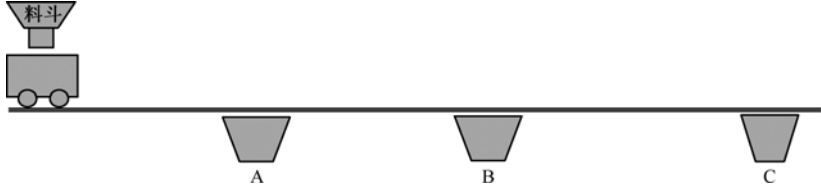
6. 动力头按图 6.73(a)所示的步骤动作：①快进，②工进 1，③工进 2，④快退。输出 M0~M3 在各步的状态如图 6.73(b)所示，表中的“1”、“0”分别表示接通和断开。设计该动力头运动的梯形图程序，要求设置手动、连续、单周期、单步 4 种工作方式。



7. 电动机拖动的运输小车，可以向 A、B、C 三个工作位运送物料，如图 6.74 所示，其动作过程如下：

- (1) 第一次，小车把物料送到 A 处并自动卸料 5 s 后返回，返回原位时料斗开关打开，装料 10 s 后，料斗开关关闭并启动第二次送料。
- (2) 第二次，小车把物料送到 B 处并自动卸料 5 s 后返回，返回原位时料斗开关打开，装料 10 s 后，料斗开关关闭并启动第三次送料。
- (3) 第三次，小车把物料送到 C 处并自动卸料 5 s 后返回，返回原位时料斗开关打开，装料 10 s 后，料斗开关关闭并启动第四次送料(物料送到 A 处)，此后重复上述送料过程。

设计一个满足要求的梯形图程序，要求有手动、单周期、连续三种工作方式，并画出 PLC 外部的接线图和操作盘的面板布置图。



8. 物品分选系统的布置如图6.75所示，传送带由电动机拖动。电动机每转过一定角度时 PH1 就发出一个脉冲(步脉冲)，对应物品在传送带上移动一定的距离。系统的控制要求如下：

(1) 传送带上的物品经过 A 处时,检测头 PH2 对物品进行检验。若属于次品,在物品到达 B 处(B 与 A 处相距 4 个步脉冲)时,电磁铁线圈接通并动作,将次品推入次品箱。次品经过光电开关 PH3 使之发出信号,记录次品个数并使电磁铁线圈断电。若 1 小时内次品超出 5 个,应发出不停机的故障报警。

(2) 若物品属于正品,则继续前行到 C 处并落入正品箱。正品经过光电开关 PH4 时使之发出信号,以记录正品的数量。

(3) 正品箱中满 100 个物品时传送带自停,并启动封箱机(液压电磁阀控制)封箱。人工搬走成品箱并更换空箱,操作限时 15 s。之后传送带又自行启动,循环上述过程。

要求:采用 CP1 系列 PLC 进行控制,作 I/O 分配,画出 PLC 的外部接线图及电动机的主电路图,并设计梯形图程序。

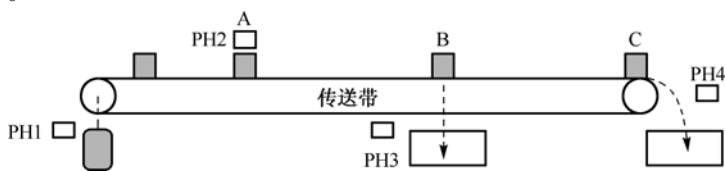


图 6.75

9. 有 4 队参加的抢答比赛,其使用设备和比赛规则做如下安排。主持人所在的总台设有一个总台灯、一个总台开始按钮、一个总台复位按钮。各队所在的分台各有一个分台灯,同时各有一个分台抢答按钮。当主持人给出题目,说“开始”并同时按下总台开始按钮后,方可进行抢答,如果哪队提前抢答,则做抢答违例处理,若过了 20 s 还无队抢答,则做“无队抢答处理”。在有队抢答时,抢得之队必须在 50 s 内完成答题,如在 50 s 内没答完(主持人不能进行复位处理),则做“答题超时”处理。

10. 如图 6.76 所示,试设计用 CP1H PLC 控制 4 层电梯的顺序控制程序。控制要求如下:

(1) 根据各楼层或梯内的呼叫次序及电梯所处的运行状态,实现电梯的启动与停止,自动判断电梯的上行和下行,原则是先响应与电梯运行的同方向呼叫,再响应反方向的呼叫。

(2) 电梯在各层停稳后,自动开门,当门开到位、触及限位开关后停止,并保持 3s,然后自动关门,当门关到位、触及限位开关后停止,此时电梯才能上行或下行。可以手动强制开门或关门。

(3) 利用数码管显示轿箱所处楼层位置;各楼层呼叫按钮带指示灯显示,轿箱到位后指示灯灭。

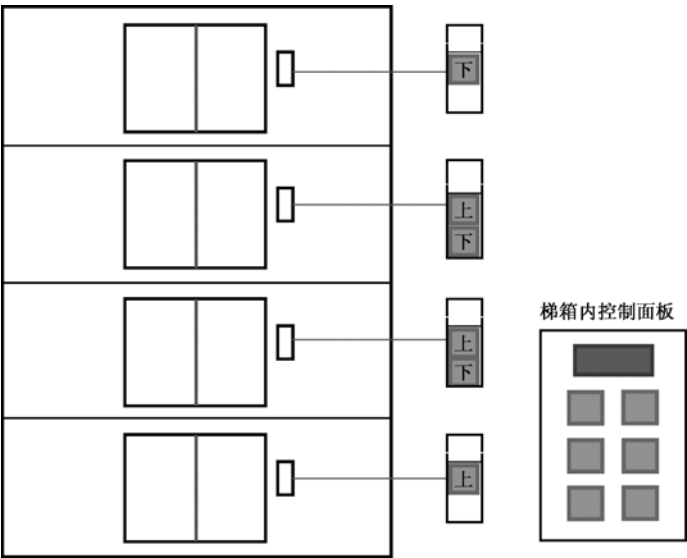


图 6.76



# 第 7 章 CP1 系列 PLC 的模拟量功能

CP1 系列 PLC 的模拟量的输入/输出可以通过两种方式实现：

- ① XA 型的 CP1H CPU 单元中内置模拟量输入 4 点/输出 2 点，NA 型的 CP1E CPU 单元中内置模拟量输入 2 点/输出 1 点。
- ② 当内置模拟量输入/输出不能满足要求或使用不含内置模拟量功能的 CP1 系列 PLC 时，采用 CP1W(或 CPM1A)模拟量扩展单元扩展模拟量输入/输出。

## 7.1 CP1 系列 PLC 的内置模拟量输入/输出功能

### 7.1.1 模拟量输入/输出功能介绍

#### 1. XA 型 CP1H 模拟量输入/输出功能

在 2.1.2 节中介绍过，XA 型 CP1H 和 NA 型 CP1E CPU 单元具有内置模拟量输入/输出功能。本节以 XA 型 CP1H PLC 为例说明此功能，XA 型的 CP1H CPU 单元中有内置模拟输入 4 点及模拟输出 2 点。内置模拟量输入/输出通过模拟量输入/输出端子台接入，而输入形式的选择通过模拟电压输入/电流输入切换开关实现，如图 7.1 所示，端子台和切换开关位于 CP1H PLC 面板的左下角。

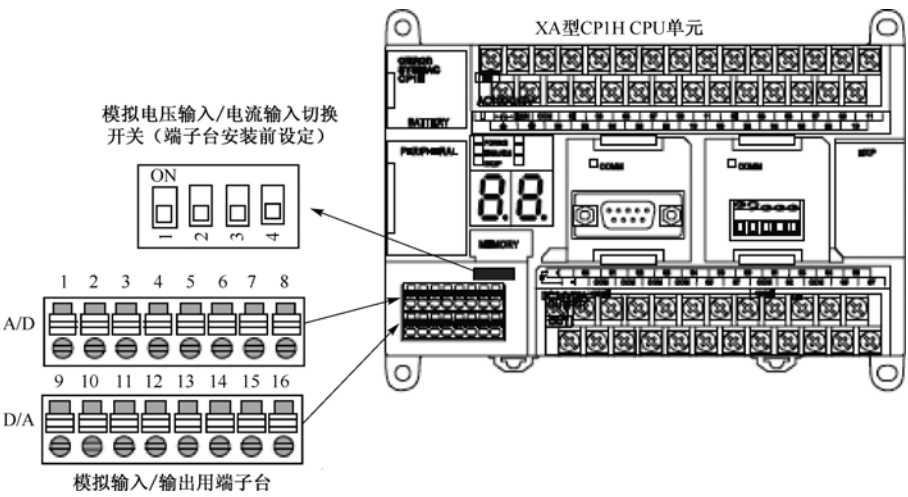


图 7.1 XA 型 CP1H 内置模拟量输入/输出设备

① 内置模拟输入切换开关(XA 型 CP1H)。将各模拟输入在电压输入下使用还是电流输入下使用间切换。模拟输入切换开关的设定见表 7.1。

表 7.1 模拟输入切换开关的设定

开关 No.	输入 No.	设 定 功 能
1	输入 1	ON: 电流输入 OFF: 电压输入 (出厂设定为电压输入)
2	输入 2	
3	输入 3	
4	输入 4	

② 内置模拟输入/输出端子台(XA 型 CP1H)。用于连接模拟输入 4 点、模拟输出 2 点的端子。各端子(引脚)的定义见表 7.2。

表 7.2 XA 型 CP1H 模拟量输入/输出引脚定义

类 型	引 脚 号	符 号	含 义
模拟输入	1	VIN0/IIN0	第 0 路模拟量电压/电流输入(接正极)
	2	COM0	第 0 路模拟量输入公共端(接负极)
	3	VIN1/IIN1	第 1 路模拟量电压/电流输入(接正极)
	4	COM1	第 1 路模拟量输入公共端(接负极)
	5	VIN2/IIN2	第 2 路模拟量电压/电流输入(接正极)
	6	COM2	第 2 路模拟量输入公共端(接负极)
	7	VIN3/IIN3	第 3 路模拟量电压/电流输入(接正极)
	8	COM3	第 3 路模拟量输入公共端(接负极)
模拟输出	9	VOUT0	第 0 路模拟量电压输出(接正极)
	10	IOUT0	第 0 路模拟量电流输出(接正极)
	11	COM0	第 0 路模拟量输出公共端(接负极)
	12	VOUT1	第 1 路模拟量电压输出(接正极)
	13	IOUT1	第 1 路模拟量电流输出(接正极)
	14	COM1	第 1 路模拟量输出公共端(接负极)
	15	AG	模拟 0V
	16	AG	模拟 0V

2. XA 型 CP1H 模拟量输入/输出功能

内置模拟量输入/输出功能见表 7.3。

表 7.3 XA 型 CP1H 内置模拟输入/输出功能

项 目		电压输入/输出 <sup>①</sup>	电流输入/输出 <sup>①</sup>
模拟输入	模拟输入点数	4 点(占用 200~203CH, 共 4CH)	
	输入信号量程	0~5 V、1~5 V、0~10 V、-10~10 V	0~20 mA、4~20 mA
	最大额定输入	± 15 V	± 30 mA
	外部输入阻抗	1 MΩ以上	约 250 Ω
	分辨率	1/6000 或 1/12000(FS: 满量程) <sup>②</sup>	
	综合精度	25℃ ± 0.3%FS/0~55℃ ± 0.6%FS	25℃ ± 0.4%FS/ 0~55℃ ± 0.8%FS
	A/D 转换数据	-10~10 V 时: 满量程值 F448(E890)~0BB8(1770) Hex 上述以外: 满量程值 0000~1770(2EE0) Hex	
	平均化处理	有(通过 PLC 系统设定来设定各输入)	
	断线检测功能	有(断线时的值 8000 Hex)	
模拟输出	模拟输出点数	2 点(占用 210CH、211CH, 共 2CH)	
	输出信号量程	0~5 V、1~5 V、0~10 V、-10~10 V	0~20 mA、4~20 mA
	外部输出允许负载电阻	1 kΩ以上	600 Ω以下
	外部输出阻抗	0.5 Ω以下	—
	分辨率	1/6000 或 1/12000(FS: 满量程) <sup>②</sup>	
	综合精度	25℃ ± 0.4%FS / 0~55℃ ± 0.8%FS	
	D/A 转换数据	-10~10 V 时: 满量程值 F448(E890)~0BB8(1770) Hex 上述以外: 满量程值 0000~1770(2EE0) Hex	
转换时间		1 ms/点 <sup>③</sup>	
隔离方式		模拟输入/输出与内部电路间: 光电耦合器隔离 (但模拟输入/输出间为不隔离)	

注: ① 电压输入/电流输入的切换由内置模拟输入切换开关来完成(出厂时设定为电压输入)。  
② 分辨率 1/6000、1/12000 的切换由 PLC 系统设定来进行, 所有的输入/输出通道只能用同一个分辨率设定。  
③ 合计转换时间为所使用的点数的转换时间的合计, 使用模拟输入 4 点+模拟输出 2 点时为 6 ms。

模拟量输入单元的功能是将标准的电压信号(−10~10 V、0~5 V、0~10 V 或 1~5 V)或电流信号(0~20 mA 或 4~20 mA)转换成数字量后送入 PLC 中的对应存储通道中。

(1) 基本技术指标

模拟量输入/输出功能的基本技术指标包括分辨率和输入/输出信号量程。

(2) 平均值处理功能(模拟输入)

在平均化处理功能中,将前 8 次输入的平均值(动态平均值)作为转换数据输出。输入发生细微变动的情况下,可通过平均化处理,保持输入平滑。通过 CX-P 的 PLC 系统设定,逐个设定到 PLC 的各模拟量输入/输出。

(3) 断线检测功能(模拟输入)

输入量程为 1~5 V、输入信号不足 0.8 V 时,或输入量程为 4~20 mA、输入信号不足 3.2 mA 时,判断为输入布线发生断线,断线检测功能工作。如断线检测功能工作,数据转为 8000 Hex。断线检测功能的工作时间、解除时间与转换时间相同。输入再次恢复到可转换的范围时,断线检测功能被自动解除,恢复到通常的转换数据。断线检测标志被分配到特殊辅助继电器 A434 CH 位 00~03。

3. NA 型 CP1E 模拟量输入/输出功能

NA 型 CP1E 内置模拟量有专门的电压输入端和电流输入端以及电压输出端和电流输出端,不需要电压电流切换开关选择电压或电流。NA 型的 CP1E PLC 输入/输出端子排列如图 7.2 所示。

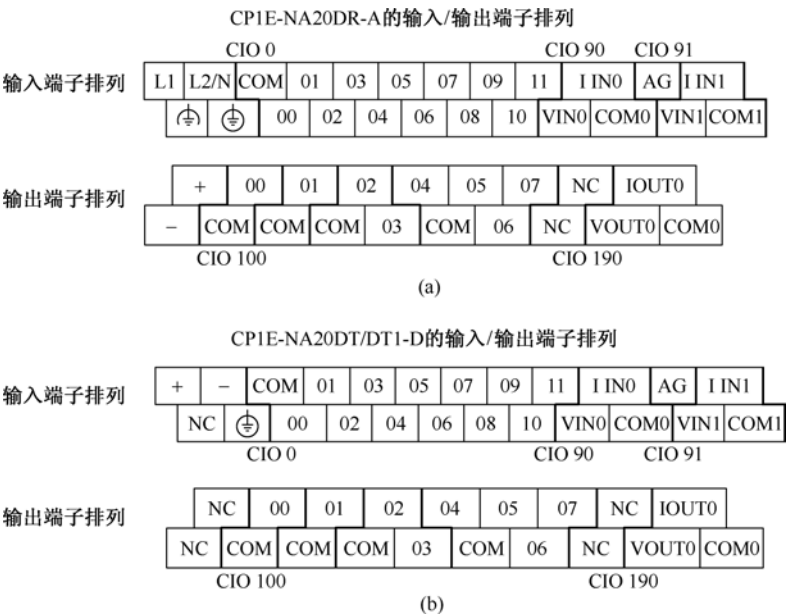


图 7.2 NA 型 CP1E PLC 的输入输出端子排列

图 7.2 中 CP1E PLC 模拟量输入/输出引脚定义见表 7.4。

NA 型的 CP1E 与 XA 型 CP1H 模拟量输入/输出的许多性质是一样的,例如输入信号量程、最大额定输入、外部输入阻抗和综合精度等,相同之处见表 7.3。这里把两者进行比较,不同之处列于表 7.5 中。

表 7.4 NA 型 CP1E 模拟量输入/输出引脚定义

类 型	符 号	含 义
模拟输入	VIN0	第 0 路模拟量电压输入
	IIN0	第 0 路模拟量电流输入
	COM0	第 0 路模拟量输入公共端
	VIN1	第 1 路模拟量电压输入
	IIN1	第 1 路模拟量电流输入
	COM1	第 1 路模拟量输入公共端
	AG	模拟 0V
模拟输出	VOUT0	第 0 路模拟量电压输出
	IOUT0	第 0 路模拟量电流输出
	COM0	第 0 路模拟量输出公共端

表 7.5 XA 型 CP1H 与 NA 型的 CP1E 内置模拟输入/输出功能比较

内容 机型	通道号		端子号	分辨率	转换时间
XA 型 CP1H	A/D	200CH	模拟输入 0	2 种：6000 和 12000	1 ms/点 模拟输入 4 点+模拟输出 2 点，共 6 ms
		201CH	模拟输入 1	6000 时：	
		202CH	模拟输入 2	-10~10 V 量程：F448~0BB8 Hex	
		203CH	模拟输入 3	其他量程：0000~1770 Hex	
	D/A	210CH	模拟输出 0	12000 时：	
		211CH	模拟输出 1	-10~10 V 量程：E890~1770 Hex 其他量程：0000~2EE0 Hex	
NA 型 CP1E	A/D	90CH	模拟输入 0	1 种：6000	2 ms/点 拟输入 2 点+模拟输出 1 点，共 6 ms
		91CH	模拟输入 1	-10~10 V 量程：F448~0BB8 Hex	
	D/A	190CH	模拟输出 0	其他量程：0000~1770 Hex	

7.1.2 模拟量输入/输出功能的使用

1. 输入切换开关设定

切换 CP1H 的各模拟输入，使其在电压输入下使用或在电流输入下使用。切换开关为 ON，电流输入；切换开关为 OFF，电压输入。

CP1H 的模拟输出有专门的电压输出端和电流输出端，不需要做选择设定。

2. PLC 系统设定

使用 CP1H 的模拟量输入/输出功能之前必须对 PLC 进行设定。先将 CX-P 编程软件与 PLC 连接，在 CX-P 的工程目录，双击“设置”，弹出“PLC 设定”对话框，在“内建 AD/DA”选项卡中进行模拟量功能的设置，如图 7.3 所示。

对 4 路输入和 2 路输出模拟量分别选择：是否使用、输入/输出量程、是否使用平均化处理。分辨率的设定是对所有的输入/输出而言的。

电压输入/输出信号范围(量程)有 4 种：0~5 V、1~5 V、0~10 V 和-10~10 V。电流输入/输出信号范围(量程)有两种：0~20 mA、4~20 mA。

当输入信号为负电压时，转换值为二进制的补码。



图 7.3 CP1H 的模拟量输入/输出功能的设定

3. 输入/输出连线

XA 型 CP1H 输入/输出连线如图 7.4 所示。

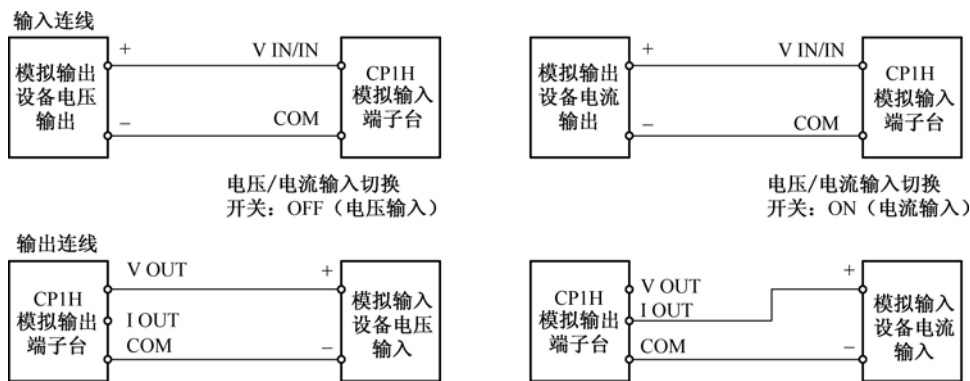


图 7.4 XA 型 CP1H 输入/输出连线

NA 型 CP1E 输入/输出连线如图 7.5 所示。

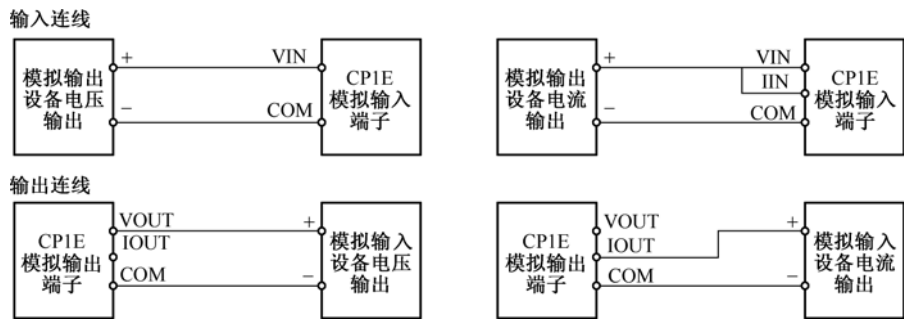


图 7.5 NA 型 CP1E 输入/输出连线

4. 内置模拟量相关的特殊寄存器区

若输入量程为 1~5 V 且输入信号不足 0.8 V (或输入量程为 4~20 mA 且输入信号不足 3.2 mA) 时，系统判断为输入断线。

由表 7.6 可知，通过断线检测功能获得的模拟输入断线信息被送到特殊辅助继电器断线检测标志 (A434 CH 位 00~03)；内置模拟输入/输出的初始处理结束信息，被送到特殊辅助继电器内置模拟初始处理结束标志 (A434 CH 位 04)中。

表 7.6 CP1H 的内置模拟量模块相关的特殊寄存器区

地 址	位	说 明	状 态
A434	00	AD0 断线异常 (CP1E 有)	内置模拟的异常发生时，为 1 (ON)
	01	AD1 断线异常 (CP1E 有)	
	02	AD2 断线异常	
	03	AD3 断线异常	
	04	内置模拟的初始处理完成标志 (CP1E 有)	内置模拟初始处理完成时，标志为 1

7.2 CP1 系列 PLC 的扩展模拟量输入/输出单元

CP1H CPU 单元可以连接 CP1W、CPM1A 系列和 CJ 系列的扩展模拟量输入/输出单元。CP1L/CP1E CPU 单元只能连接 CP1W 和 CPM1A 系列的扩展模拟量输入/输出单元。本书只介绍 CP1H CPU 单元连接 CPM1A 系列的扩展模拟量输入/输出单元。

7.2.1 CP1W 和 CPM1A 系列模拟量扩展单元

CP1H 能够连接 CPM1A 系列的扩展单元，能连接 CPM1A 扩展 I/O 单元最多为 7 台。但是，温度调节单元 CPM1A-TS002/102 中因为占有输入继电器区域 4CH，当包含这些单元时，要减少可连接的台数，通道的分配如图 7.6 所示。

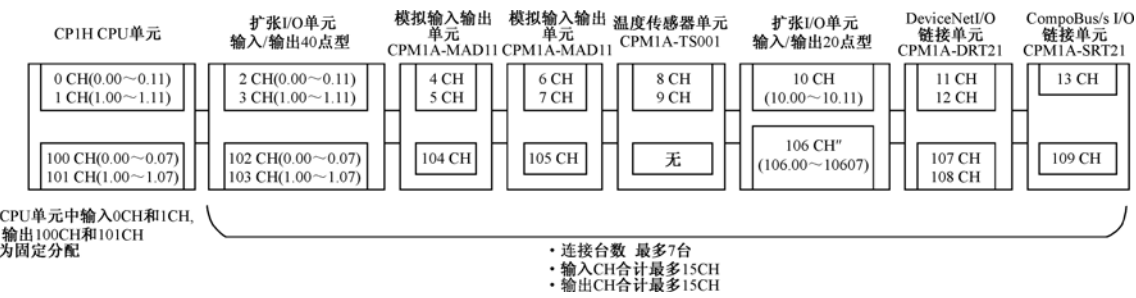


图 7.6 CP1H 系列的扩展单元的通道分配

CP1 系列 PLC 能连接 CPM1A 或 CP1W 的扩展单元，见表 7.7，最多台数为 7 台。CP1 系列 PLC 的模拟量扩展单元有多种，下面以 CPM1A-MAD01 为例，介绍模拟量扩展单元。

表 7.7 CP1W(或 CPM1A) 的特殊 I/O 扩展单元

名 称	型 号	规 格			占用通道数
模拟量输入单元	CP1W-AD041 CPM1A-AD041	模拟输入：4 点	电压：0~5 V/1~5 V/0~10 V/ -10~+10 V 电流：0~20 mA/4~20 mA	分辨率 6000	输入 4/输出无
模拟量输出单元	CP1W-DA041 CPM1A-DA041	模拟输出：4 点	电压：1~5 V/0~10 V/-10~+10 V 电流：0~20 mA/4~20 mA	分辨率 6000	输入无/输出 4

(续表)

名 称	型 号	规 格			占用通道数
模拟量 I/O 单元	CPM1A-MAD01	模拟输入：2 点	电压：0~10 V/1~5 V 电流：4~20 mA	分辨率 256	输入 2/输出 1
		模拟输出：1 点	电压：0~10 V/-10~+10 V 电流：4~20 mA		
	CP1W-MAD11 CPM1A-MAD11	模拟输入：2 点	电压：0~5 V/1~5 V/ 0~10 V/-10~+10 V 电流：0~20 mA/4~20 mA	分辨率 6000	
		模拟输出：1 点	电压：1~5 V/0~10 V/-10~+10 V 电流：0~20 mA/4~20 mA		
温度传感器单元	CP1W-TS001 CPM1A-TS001	输入：2 点	热电偶输入 K、J 之间选一		输入 2/输出无
	CP1W-TS002 CPM1A-TS002	输入：4 点			输入 4/输出无
	CP1W-TS101 CPM1A-TS101	输入：2 点	铂热电阻输入 Pt100、JPt100 之间选一		输入 2/输出无
	CP1W-TS102 CPM1A-TS102	输入：4 点			输入 4/输出无
DeviceNet I/O 链接单元	CPM1A-DRT21	DeviceNet 从站 I/O 点数：输入 32 点，输出 32 点			输入 2/输出 2
CompoBus/S I/O 链接单元	CP1W-SRT21 CPM1A-SRT21	CompoBus/S 从站 I/O 点数：输入 8 点，输出 8 点			输入 1/输出 1

7.2.2 CPM1A -MAD01 模拟量 I/O 单元

1. CPM1A-MAD01 模拟量 I/O 单元

CPM1A-MAD01 模拟量 I/O 单元有模拟输入 2 点、输出 1 点，CPM1A-MAD01 模拟量 I/O 单元如图 7.7 所示。

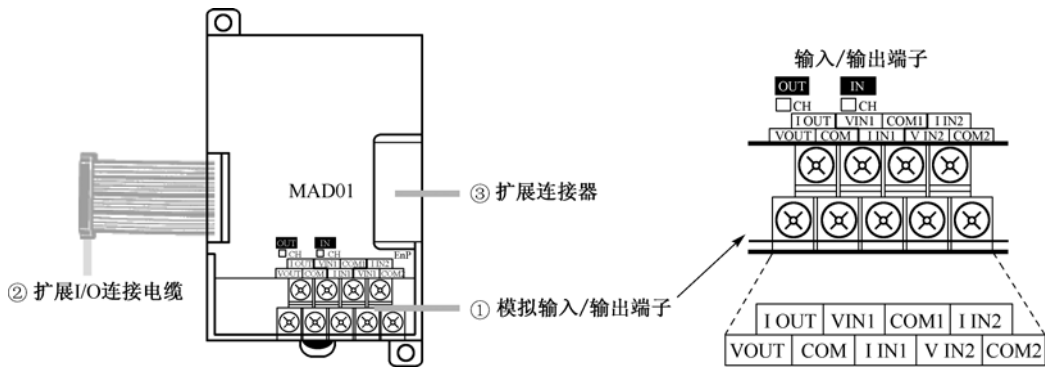


图 7.7 CPM1A-MAD01 模拟量 I/O 单元

- ① CPM1A-MAD01 模拟输入/输出端子的含义见表 7.8。注意，当使用电流输入时，须将 V IN1 和 I IN1、V IN2 和 I IN2 短接。
- ② CPM1A-MAD01 模拟输入/输出单元的性能指标见表 7.9。

表 7.8 CPM1A-MAD01 模拟输入/输出端子的含义

符 号	名 称
V OUT	电压输出
I OUT	电流输出
COM	输出 COM
V IN1	电压输入 1
I IN1	电流输入 1
COM1	输入 COM1
V IN2	电压输入 2
I IN2	电流输入 2
COM2	输入 COM2

表 7.9 CPM1A-MAD01 模拟输入/输出单元的性能指标

项 目		电压输入/输出	电流输入/输出
模拟输入	模拟输入点数	2 点	
	输入信号量程	0~10 V、1~5 V	4~20 mA
	最大额定输入	± 15 V	± 30 mA
	外部输入阻抗	1 MΩ以上	约 250 Ω
	分辨率	1/256	
	综合精度	1.0%FS (FS: 满量程)	25℃ ± 0.4%FS/0~55℃ ± 0.8%FS
	A/D 转换数据	8 位二进制	
	断线检测功能	1~5 V/4~20 mA 量程时有	
模拟输出	模拟输出点数	1 点	
	输出信号量程	0~10 V、-10~10 V	4~20 mA
	外部输出允许负载电阻	5 Ma	—
	外部输出阻抗	—	350 Ω
	分辨率	1/256 (输出信号量程-10~10 V 时为 1/512)	
	综合精度	1.0%FS (FS: 满量程)	
	D/A 转换数据	8 位二进制+符号位	
转换时间		10 ms/单元	
隔离方式		模拟输入/输出端子与 PLC 信号间：光电耦合器隔离 (模拟输入/输出信号间不隔离)	

2. CPM1A-MAD01 模拟量 I/O 单元与 CP1H CPU 单元主机的连接

CPM1A-MAD01 模拟量 I/O 单元与 CP1H CPU 单元主机的连接如图7.8所示。

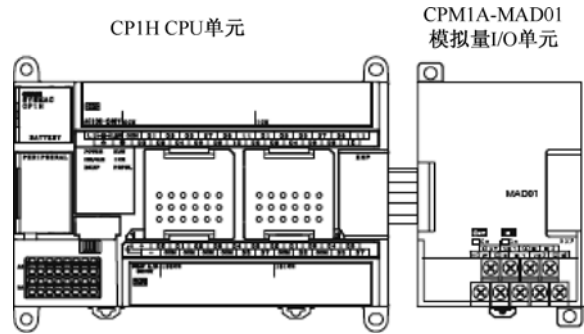


图 7.8 CPM1A-MAD01 模拟量 I/O 单元与 CP1H CPU 单元主机的连接图



3. CPM1A-MAD01 模拟量单元的布线

电压/电流的选择用布线来切换，模拟输入布线如图7.9所示。注意电压电流输入的不同连接。

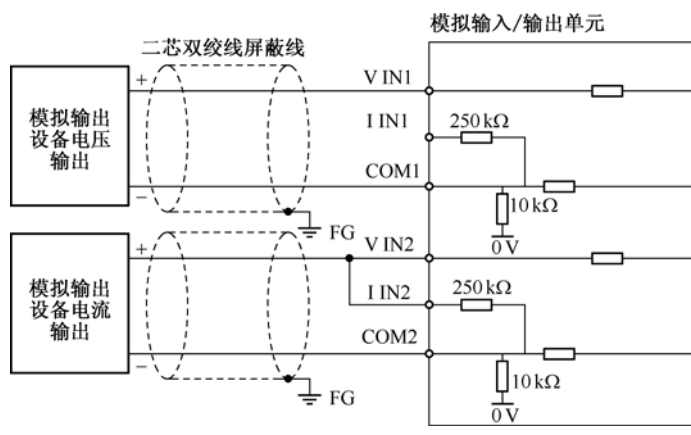


图 7.9 CPM1A-MAD01 模拟输入布线

模拟输出布线如图7.10所示。注意电压电流输出的不同连接。

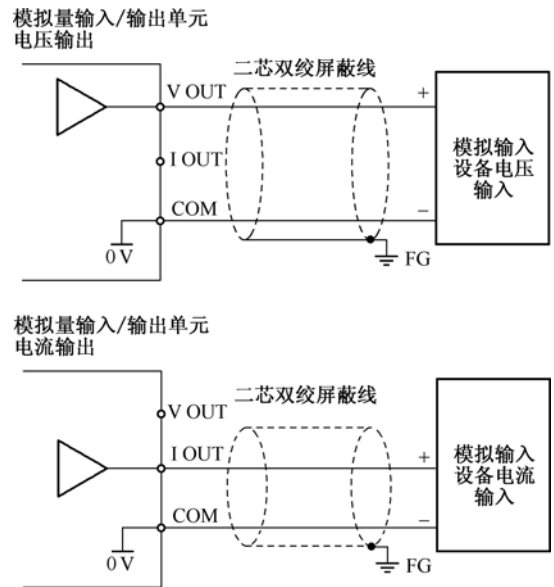


图 7.10 CPM1A-MAD01 模拟输出布线

注意，模拟输出可以将电压输出和电流输出同时使用，但是输出的电流合计请控制在 21 mA 以下。

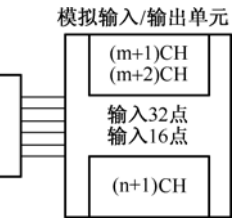


图 7.11 CPM1A-MAD01 模拟量单元的通道分配

4. CPM1A-MAD01 模拟量单元输入/输出继电器的分配

按与 CPU 单元连接的扩展单元的顺序分配通道，给 CPM1A-MAD01 分配输入 2 CH 和输出 1 CH。若前面单元的最后的输入/输出通道分别为 m 和 n,则 CPM1A-MAD01 扩展单元的通道分配如图7.11 所示。

5. CPM1A-MAD01 量程代码的写入

只有将量程代码写入到 (n+1)CH, A/D、D/A 才能开始转换。有 8 种量程代码 FF00~FF07, 见表 7.10。这也就意味着 (n+1)CH 通道即是将量程代码写入的通道, 又是模拟输出通道。

表 7.10 量程代码的含义

量 程 代 码	模拟输入 1 信号量程	模拟输入 2 信号量程	模拟输出信号量程
FF00	0~10 V	0~10 V	0~10 V/4~20 mA
FF01	0~10 V	0~10 V	-10~+10 V/4~20 mA
FF02	1~5 V/4~20 mA	0~10 V	0~10 V/4~20 mA
FF03	1~5 V/4~20 mA	0~10 V	-10~+10 V/4~20 mA
FF04	0~10 V	1~5 V/4~20 mA	0~10 V/4~20 mA
FF05	0~10 V	1~5 V/4~20 mA	-10~+10 V/4~20 mA
FF06	1~5 V/4~20 mA	1~5 V/4~20 mA	0~10 V/4~20 mA
FF07	1~5 V/4~20 mA	1~5 V/4~20 mA	-10~+10 V/4~20 mA

注: ① 量程代码在运行开始时写入到 (n+1)CH, 到写入量程代码前模拟输入/输出的转换不会开始。  
② 量程代码一旦设定好, 电源在 ON 时量程代码不能变更。要变更输入/输出量程时, 必须重新接通电源。  
③ 写入上面表格以外的量程代码时, 模拟输入/输出不接受设定。另外, A/D、D/A 转换也不能开始。

例 7-2-1 量程代码的写入。

操作数 1 #FF02 为量程代码, 表明模拟输入 1 量程为 1~5 V/4~20 mA; 模拟输入 2 量程为 0~10 V; 模拟输出量程为 0~10 V/4~20 mA。操作数 2 的 n+1 为量程代码写入通道, 也是模拟量输出通道。梯形图如图 7.12 所示。

6. 读出 A/D 转换数据

A/D 转换数据输出在 (m+1)CH 和 (m+2)CH 的 00~07 位, 如图 7.13 所示。

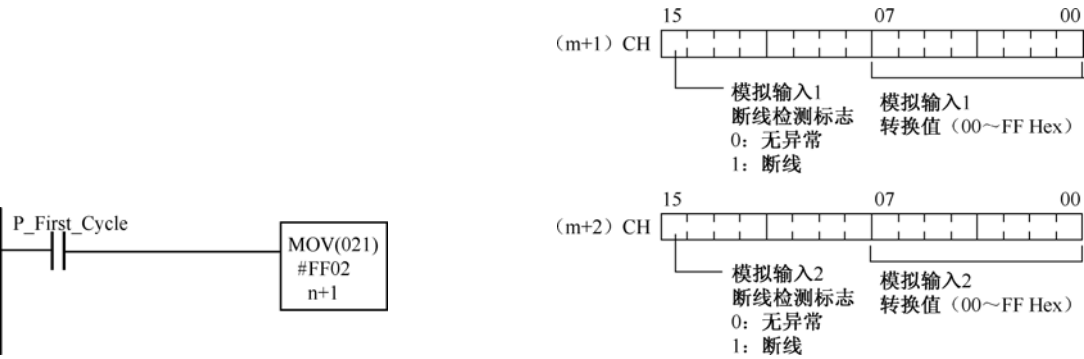


图 7.12 例 7-2-1 梯形图

图 7.13 A/D 转换数据输出

断线检测标志在 1~5 V/4~20 mA 输入信号量程时, 输入信号电压/电流为 1 V 或者 4 mA 以下为 ON。在 0~10 V 输入信号量程内不能使用。

7. D/A 转换数据的设定

向分配给模拟输入/输出单元的输出口 (n+1)CH 写入输出数据。(n+1)CH 用于电源刚通电时量程代码的写入, 又是正常工作时的模拟量输出通道。做模拟量输出通道的意义如图 7.14 所示。

注意:

- ① 0~10 V/4~20 mA 输出信号量程范围时, 设定值范围为 0000~00FF。

②  $-10\sim+10\text{ V}$  输出信号量程范围时, 正数设定值范围为  $0000\sim00\text{FF}$ , 负数设定值范围为  $8000\sim80\text{FF}$ 。

③ 输出值设定时,  $-10\sim+10\text{ V}$  输出信号时 8~14 bit 无用,  $0\sim10\text{ V}/4\sim20\text{ mA}$  输出信号时 8~15 位无用。



图 7.14 D/A 转换数据输出

8. 电源为 ON 时的模拟输入/输出等待程序

接通电源, 从开始运行后到最初的转换数据保存到输入通道中为止, 要花费两个周期  $100\text{ ms}$  左右的时间。因此在电源为 ON 时开始运行时, 同时应制订如例 7-2-2 所示的有效转换数据等待程序。在初始处理完成前, 模拟输入数据为  $0000$ 。模拟输出在写入量程代码之前为  $0\text{ V}$  或者  $0\text{ mA}$ , 量程代码写入后,  $0\sim10\text{ V}/-10\sim+10\text{ V}$  量程时的输出为  $0\text{ V}$ ,  $4\sim20\text{ mA}$  量程时的输出为  $4\text{ mA}$ 。

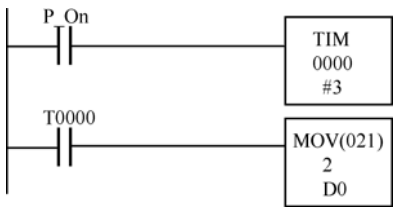


图 7.15 例 7-2-2 梯形图

例 7-2-2 模拟输入/输出等待程序。

电源为 ON 时  $\text{TIM}0000$  启动。经过  $0.2\sim0.3\text{ s}$  ( $200\sim300\text{ ms}$ ),  $\text{TIM}0000$  的触点  $\text{T}0000$  也转为 ON, 保存在  $2\text{CH}$  里的模拟输入 0 的转换数据向  $\text{D}0$  传送。梯形图如图 7.15 所示。

9. 模拟输入/输出单元异常时的处理方法

模拟输入/输出发生异常时, 模拟输入数据为  $0000$ 、模拟输出为  $0\text{ V}$  或者  $4\text{ mA}$ 。CPM1A 系列扩展单元的异常输出到  $\text{A}436\text{ CH}$  (位  $0\sim6$ )。从靠近 CPU 单元开始从  $\text{A}436.00$  按顺序分配。请在要用梯形图程序检测出扩展单元的异常时使用。

例 7-2-3 模拟输入 0 量程为  $0\sim10\text{ V}$ , 模拟输入 1 量程为  $1\sim5\text{ V}/4\sim20\text{ mA}$ , 模拟输出量程为  $0\sim10\text{ V}/4\sim20\text{ mA}$ 。梯形图如图 7.16 所示。

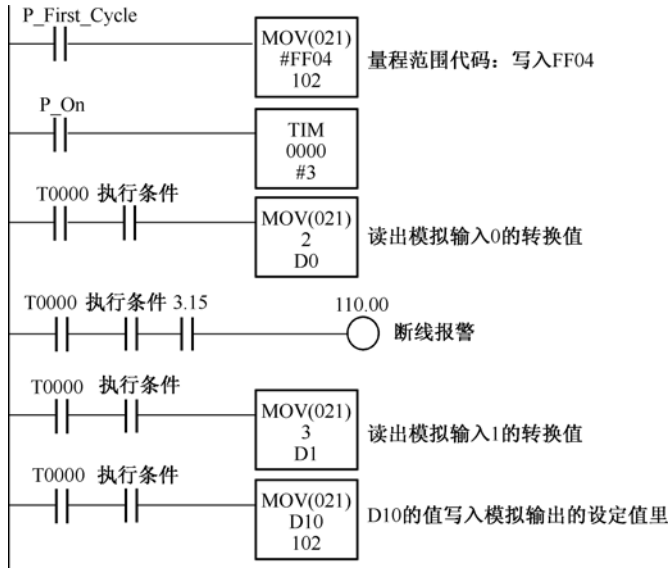


图 7.16 例 7-2-3 梯形图

## 习 题

1. 使用 CP1H 的内置模拟量输入/输出功能时，需做哪些设定？使用 CP1H 的模拟量 I/O 单元时，需做哪些设定？
2. 假设已用 CP1H 的内置模拟量输入 0 采集到被控变量。设计程序实现：把被控模拟量的实际值与设定值模拟量输入(R)进行比较，按照比较结果，产生相应的 ON 或 OFF 的继电控制输出。
3. 用 CP1H 的内置模拟量输入输出功能采集系统的被控变量，对系统进行控制。设计一个简单反馈控制程序，要求：用设定值(R)与实际反馈值(C)之差(E)进行控制。E 经放大(乘 H)后，产生控制输出 U，作用于被控对象。
4. 对一个简单反馈控制系统，采用 CP1H 的内置模拟量输入输出功能采集系统的被控变量，对系统进行控制。试编写 PID 控制程序。

# 第 8 章 欧姆龙 3G3MX2 变频器

## 8.1 变频器概述

变频器的英文是 VFI (Variable-Frequency Inverter)，变频器是应用变频技术与微电子技术，通过改变电动机工作电源电压的频率和幅度的方式来控制交流电动机转速的电力传动元件，通过它达到平滑控制电动机转速的目的。通常指将频率固定的工频交流电，变换成频率可调的三相交流电的电力控制装置。变频器的出现，使复杂的调速控制简单化，用变频器+交流鼠笼式感应电动机组合替代了大部分原先只能用直流电动机完成的工作，缩小了体积，降低了维修率，使传动技术发展到了新的阶段。

### 8.1.1 变频器的构成

从结构上看，变频器分为间接变频和直接变频两大类。间接变频器先将工频交流电源(整流)变换为直流，再经过逆变器将直流电变换为电压和频率可控的交流电，简称为“交-直-交”变频器。直接变频器将工频交流电直接变换为电压和频率可控的交流电，不需要中间整流环节，简称为“交-交”变频器。目前在中小容量的变频器中，“交-直-交”变频器应用最为广泛，它的基本结构如图8.1所示。“交-直-交”变频器由主电路以及控制电路两部分组成，主电路又包括整流电路、中间直流电路和逆变电路。

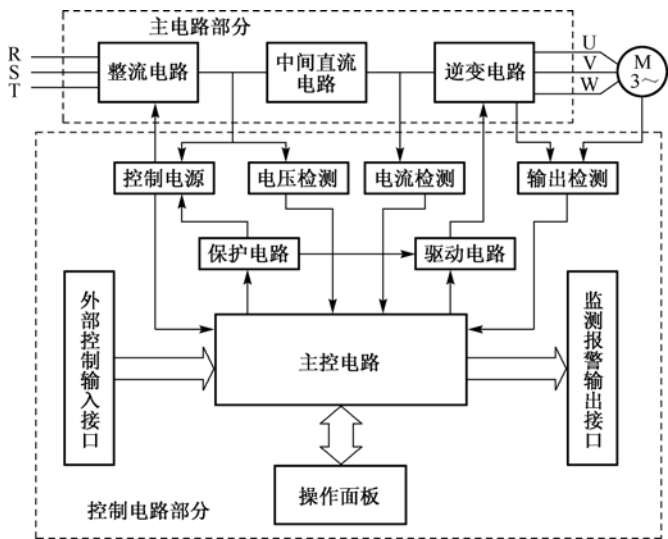


图 8.1 “交-直-交”通用变频器的硬件结构示意图

### 1. 变频器的主电路

变频器的主电路主要由整流电路、中间直流电路和逆变电路组成。如图8.2所示为电压型变频器主电路。整流电路通常由三相全波整流桥组成，用于将电网的交流电源变换为所需要的直流电源。对于电流型变频器，它提供直流电流源；对于电压型变频器，它提供直流电压源。整流元件常用二极管和

晶闸管。二极管整流电路主要用于 PWM 变频器中，如图8.2中的 VD1~VD6。输出的直流电压取决于电源电压的幅值；由晶闸管构成的整流电路，其直流电压可控。

中间直流电路主要有两个作用。一是滤波，使输出的直流电源平滑。电压型变频器采用大容量电解电容，如图8.2中的  $C_{F1}$  和  $R_{C1}$  以及  $C_{F2}$  和  $R_{C2}$ 。二是限流和短路保护，如图8.2中的限流电阻  $R_S$  和断路开关  $S_S$  等。此外，由于电动机制动的需要，在中间直流电路中，有时还包括制动电阻和其他辅助电路。有的变频器还提供交流电动机反馈制动时的再生电流通路。

逆变电路主要是在控制电路的作用下，将平滑的直流电源交换为频率和电压可调的交流电源。逆变器的输出通常就是变频器的输出，它被用来实现对异步电动机的调速控制。在图8.2中，绝缘栅双极型晶体管 V1~V6 构成逆变桥，在控制信号作用下将直流电压  $U_D$  “逆变” 成三相交流电 (U、V、W)；VD7~VD12 构成续流电路。

一般还包括交流输出电路，如输出滤波电路、驱动电路及反馈电路等。

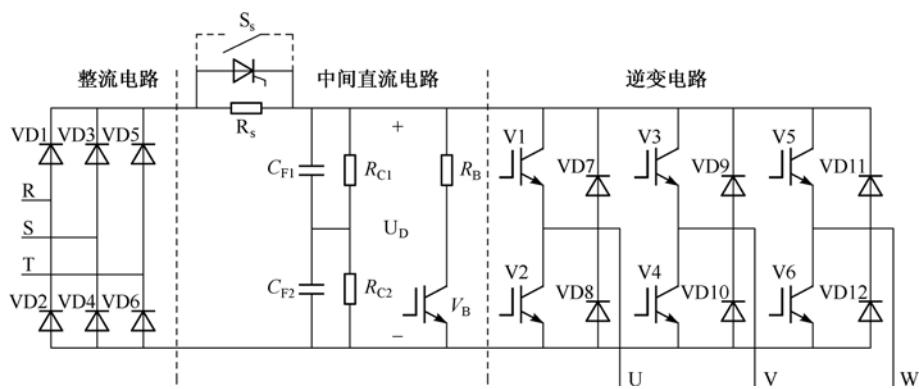


图 8.2 电压型变频器的典型主电路

2. 变频器的控制电路

控制电路的功能是按要求产生和调节一系列的控制脉冲来控制逆变器开关管的导通和关断，从而配合逆变电路完成逆变任务。变频器典型的控制电路包括主控电路、信号检测电路、(门极)驱动电路、外部接口电路、保护电路以及操作面板等部分，如图8.1所示。它和逆变器的地位都很重要，是变频器的核心部分。控制电路的性能指标及其功能，决定着变频器产品的优劣。

变频器主控电路的中心是一个高性能的微处理器，并配以 ASIC、EPROM、RAM 芯片等电路，主要有 3 种任务：

- ① 接受各种信号，其中包括各种功能的预置信号，从键盘或外接输入端输入的给定信号，外部输入端送来的控制信号以及从各种检测电路取样的状态信号等。
- ② 进行基本运算，包括进行矢量运算、实时地计算 SPWM 波形各切换点的时刻以及其他运算等。
- ③ 输出计算结果，包括通过驱动电路对逆变器进行波形变换，显示当前的各种状态，以及输出各种信号到外接端子上，便于测试和调整等。

除此之外，主控电路还可以实现各项控制功能和实施各项保护功能，例如，对 SPWM 信号进行启动、停止、升速、降速、点动等控制和故障报警、紧急终止输出等。

电压、电流和电动机输出速度检测电路主要作用是将变频器主电路和电动机的工作状态反馈至微处理器，并由它按照预先确定的算法进行处理后，为各部分电路提供控制信号和保护信号，以控制变频器输出和为变频器及电动机提供必要的保护。

保护电路包括输入和输出的过压保护、欠压保护、过载保护、过流保护、短路保护、过热保护等。

在不少应用场合,变频器自身还有过速保护、失速保护、制动控制等辅助电路。主要作用是由微处理器将变频器对检测电路得到的各种信号进行算法处理,判断变频器本身或系统是否出现异常,以便进行必要的处理,包括停止变频器的输出,从而对变频器以及系统提供相应的保护。

输入和输出接口电路通常包括顺序控制指令的输入电路(主要是电动机正反转和多段调速控制信号)、频率指令(模拟信号,经 A/D 处理)输入电路、监视信号输出电路以及通信接口电路等。变频器一般都有 RS-232、RS-485/422 通信接口,以便于变频器与计算机、变频器与 PLC 或网络通信的连接。

操作面板通常包括操作键盘、数据显示器、运行状态指示 LED 等部分。操作键盘用于向主控电路发出各种指令或控制信号,数据显示器则是对主控电路提供的各种数据进行显示,状态指示用于监视变频器的运行状况。

驱动电路为逆变器的绝缘栅双极型晶体管 V1~V6 提供驱动信号。

控制电源为所有电路部分提供电源。

## 8.1.2 变频器的分类

变频器的分类方法很多,除了上述按变换环节分为“交-直-交”类型和“交-交”类型之外,常见还有以下的几种分类方法。

### 1. 按直流环节的储能方式分类

当变频器为交流电动机负载时,根据负载与中间直流电路之间无功功率的交换方式,决定着中间环节中选择不同的储能元件。据此把变频器分为电压型和电流型两类。

电压型变频器的中间直流电路采用电容元件,则直流电源的内阻抗很小,具有恒压源特性。电动机的转速精度取决于逆变频率的精度及电动机的转差率,受负载电流变化的影响小,所以电压型变频器可以采用开环控制,即通过改变给定信号,经过控制环节来实现调速。若要求高精度调速,则应采用闭环控制,即在开环基础上增加速度等(负)反馈环节,使系统可以很快地消除各种干扰,电动机的转速不随外界的扰动而变化,以精确地保持在给定的数值上。因此,通用变频器大多采用电压型。但由于电压型变频器存在较大的滤波电容,使其功率因数较低,动态响应较慢,且当电动机处于再生(制动)发电时,反馈到电容上的电能难以送入交流电网,故它适合于不频繁起动、能耗制动以及对快速性要求不高的场合。

电流型变频器的中间直流电路中串联了一个大电感,则电源的内阻抗很高,具有恒流源特性,可以限制输出电流以及吸收无功功率,受负载电压变化的影响小。当负载为电动机时,容易实现再生制动,可将反馈到直流侧的再生电能直接送入交流电网。它适合于大、中功率的传动装置和要求频繁起动、急加减速、动态响应性能要求高的大容量电动机的单机拖动场合,但它的逆变范围稍窄,不能空载工作。

### 2. 按逆变环节的调制方式分类

根据改变逆变器输出电压大小(调制)的不同方式,变频器分为脉冲幅度调制(Pulse Amplitude Modulation, PAM)和脉冲宽度调制(Pulse Width Modulation, PWM)两类。

PAM 变频器输出的交流电压是通过改变直流电压的大小,使逆变器输出脉冲的幅度发生改变来实现的。这种变频器要同时对整流电路和逆变器进行控制,其控制电路复杂,且低速运行时转速波动大,故目前的变频器几乎不再采用此类。

PWM 变频器输出的交流电压是通过改变逆变器输出脉冲的占空比来实现的。目前普遍应用的是占空比按正弦规律排列的正弦波脉宽调制(SPWM)方式,使输出电压的平均值接近正弦波,大大削弱谐波成分的影响,使调速控制得到平稳变化。

### 3. 按控制方式分类

按控制方式的不同,变频器分为 V/F(或称 U/F)控制方式、转差频率控制方式和矢量控制方式三类。

变频器还可以按逆变器主开关的器件分为普通晶闸管(SCR)、可关断晶闸管(GTO)、绝缘栅双极型晶体管(IGBT)、功率集成电路(PIC)等类型。

### 8.1.3 变频器的控制方式

变频器控制方式是针对电动机的自身特性、负载特性以及运转速度的要求,控制变频器的输出电压和频率。从而达到控制电动机转速的目的。

#### 1. V/F 控制方式

从异步电动机的特性: 转速  $n = (1-s)n_0 = \frac{60f_1}{p}(1-s)$ 、磁通  $\Phi_m \approx \frac{U_1}{4.44N_1f_1}$ 、转矩  $T = K\Phi_m I_2 \cos\varphi_2$  可知,

当改变电源频率  $f_1$  时,可改变旋转磁场的转速  $n_0$ ,也就改变了转子的转速  $n$ 。理论上  $n$  与  $f_1$  成正比,则通过改变电源频率就可以实现电动机的调速。但在维持定子电压  $U_1$  不变时,随着  $f_1$  的升高,电动机的磁通  $\Phi_m$  减小,导致  $T$  下降,最大转矩  $T_m$  也降低,严重时将使电动机堵转(停车);当减少  $f_1$  时,则使  $\Phi_m$  增加,导致电动机的磁路过饱和,励磁电流增加,铁心过热。为此变频器在逆变过程中,在改变频率  $f_1$  的同时必须按一定比例改变  $U_1$ 。例如  $f_1$  升高,则  $U_1$  随之升高,以避免出现上述问题。这种控制方式用 VVVF (Variable Voltage Variable Frequency) 表示,简称 V/F 控制,其目的是维持  $\Phi_m$  不变。

在基频 ( $f=50\text{ Hz}$ ) 以下调速时,由于  $U_1$  和  $f$  都小,定子绕组的感抗也小,使定子内阻的压降相对增加,则电动机转矩不仅无法维持  $T_m$ ,反而下降。为此,必须通过检测定子电流来适当提高  $U_1$ ,以补偿定子的电压损失来维持  $T_m$ ,所以这种调速近似为恒转矩调速。

在基频以上调速时,由于电动机受到定子绕组绝缘强度的限制,  $U_1$  不允许超过  $U_N$  (额定电压),所以  $\Phi_m$  随着  $f$  的升高(保持  $U_1 = U_N$ ) 反而下降,导致转矩  $T$  减小。同时,电动机的同步机械角速度  $\Omega$  ( $\Omega = 2\pi f / P$ ) 随  $f$  的升高而增加。由电动机的功率  $P = T\Omega$  ( $T$  和  $\Omega$  的变化方向相反)可知,这种调速近似为恒功率调速。

V/F(或 U/F)控制方式的基本特点是通过使 U/F 的值保持一定而得到所需的转矩特性,属于开环控制,电路成本低,多用于精度要求不高的通用变频器。若电动机需要低于额定转速运行时,可采用基频 ( $f=50\text{ Hz}$ ) 以下的恒转矩调速。若电动机需要高于额定转速运行时,应采用基频以上的恒功率调速,类似于直流电动机的弱磁调速情况。V/F 控制的特性曲线如图8.3所示。

V/F 控制方式如图8.4所示。主电路的逆变器用 PWM 控制。控制电路中的  $U/f$  曲线发生器根据  $f$  的设定值来确定  $U/f$  的比例关系。变频器提供有多种的  $U/f$  函数曲线,用户可根据电动机的负载性质和运行状况加以设定。

#### 2. 转差频率控制方式

转差频率控制方式是对 V/F 控制的一种改进。因为在 V/F 控制方式下,转速会随着负载的变化而改变,其变化量与转差率成正比。为了提高调速精度,就需要控制转差率。通常是用速度传感器检测电动机的运行速度,以求出转差角频率,再把它与  $f$  设定值叠加以得到新的逆变器的频率设定值  $f^*$ ,实现转差补偿。其简化的原理框图如图8.5所示。对应于频率设定值  $f$ ,经过转差补偿后定子频率的实际值为  $f^* = f + \Delta f$ ,以此进行调速控制,同时一定程度上对输出转矩进行控制,所以它与 V/F 控制方式相比,在负载发生较大变化时,仍能达到较高的速度精度和较好的转矩特性。但是,由于转差频率控制方式属于闭环控制,需要在电动机上安装速度检测器,并需要根据电动机的特性调节转差,通



常用于厂家指定的专用电动机，故通用性较差。

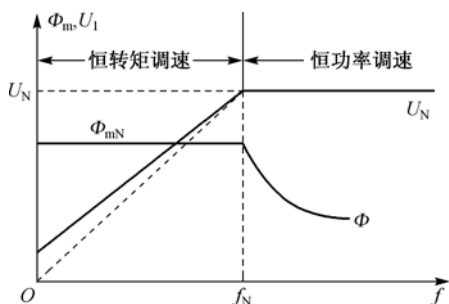


图 8.3 V/F 控制方式的特性曲线

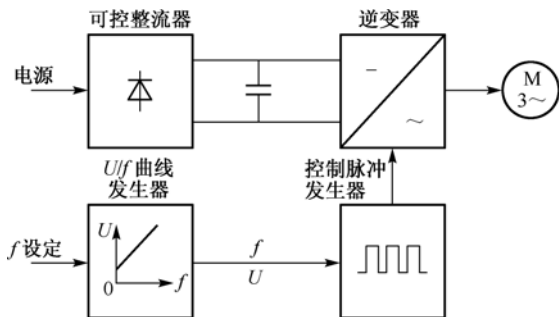


图 8.4 V/F 控制方式示意图

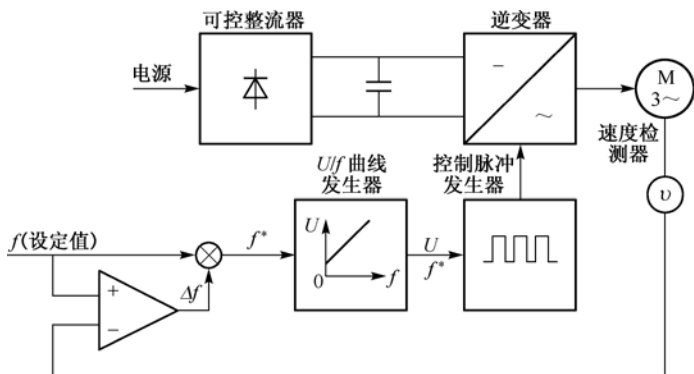


图 8.5 转差频率控制方式示意图

### 3. 矢量控制方式

上述两种控制方式都存在着动态性能指标不高的问题，而且交流异步电动机是一个多变量、强耦合、非线性的时变参数系统，很难直接通过外加(设定)信号来准确控制其电磁转矩。20 世纪 70 年代，德国 F.Blaschke 等人提出了矢量控制理论，基本出发点是以转子磁通这一旋转的空间矢量为参考坐标，利用从静止坐标系到旋转坐标系的变换，将定子电流分为产生磁场的励磁电流和与其相垂直产生转矩的转矩电流，并分别加以控制。在这种控制方式中，必须同时控制异步电动机定子电流的幅值和相位，即控制定子电流的矢量，故称为矢量控制。它通过坐标变换和重建方法，把交流电动机模型等效为直流电动机，以获得像直流电动机一样的动态性能。

图 8.6 为矢量控制方式示意图，它是在 VVVF 变频器的基础上增加了坐标变换及控制电路。控制器将给定信号分解成两个互相垂直且独立的直流信号  $i_M^*$  和  $i_T^*$  (表示励磁电流和电枢电流)。然后通过“直/交变换”将  $i_M^*$  和  $i_T^*$  变换成两相交流电流信号  $i_a^*$  和  $i_b^*$ 。再经过“2/3 变换”，将两相交流系统变换为三相交流系统，以得到三相交流控制信号  $i_A^*$ 、 $i_B^*$  和  $i_C^*$ ，去控制逆变器。对于电动机在运行过程中的三相交流系统的数据，又可以等效变换成两个相互垂直的直流信号，反馈到控制器，用于修正基本控制信号  $i_M^*$  和  $i_T^*$ 。反馈信号包括电流和速度等信息，所以它属于闭环控制。其中的电流反馈用于反映负载的状况，使直流信号中的电枢分量  $i_T^*$  能够跟随负载变化，从而模拟出类似于直流电动机的工作情况。速度反馈用于反映拖动系统的实际转速和给定值之间的差异，并进行快速校正，以提高系统的动态性能。

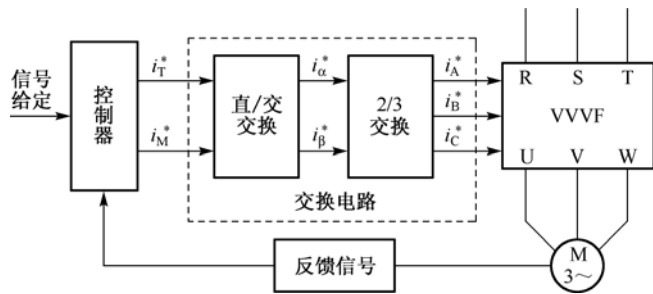


图 8.6 矢量控制方式示意图

8.1.4 变频器的主要功能

为了保证其通用性，变频器的功能较多。它除了保证自身的基本控制功能外，大多数功能是根据变频器调速系统的需要而设计的。下面按其用途，将变频器的主要功能进行分类，见表 8.1。

表 8.1 变频器的主要功能

类 型	用 途	类 型	用 途
调速控制的必要功能	自动转矩提升	与频率指令相关的功能	多段速运行
	防失速功能		频率上下线限制
	过转矩设定运行		特定频率运行禁止功能(跳跃频率)
	转差补偿		频率指令消失后的自动连续运行
	降低机械振动冲击		频率指令的反转
	运行状态的检测信号		加、减速禁止功能
	2 线和 3 线控制		加、减速时间切换功能
外部接口功能	多功能输入端	与运行方式相关的功能	S 形加、减速功能
	多功能模拟量输入端		直流制动停机(DC 制动)
	外部控制停机的信号端		运行前的直流制动
	数字信号的输入与输出端		转速搜索功能
	与上位机的通信接口		瞬时停电再启动
保护功能	电子热保护		工频电源与变频器之间的切换运行
	故障后再启动		节能运行
	制动电阻保护		多种 U/f 曲线选择功能
其他功能	载波频率设定	监视功能	负载的速度、电压、电流等参数显示
	高载波频率运行		脉冲监视功能
	平稳运行		操作面板的数据监视功能

另外，现代的变频器都具有通信功能，通过欧姆龙 CP1 系列 PLC 的串行通信接口与端口插件 CPIW-CIF11(RS422/485 口插件)与欧姆龙变频器(3G3MX2、3G3JV、3G3MV 和 3G3RV)相连，进行简单的设置后可以实现它们之间的 Modbus-RTU 通信。

8.2 欧姆龙 3G3MX2 变频器

8.2.1 欧姆龙变频器简介

欧姆龙变频器大概可分为 7 个系列：3G3JV、3G3JZ、3G3MZ-ZV2、3G3MV、3G3RV-ZV1、3G3RX 和 3G3MX2，见表 8.2。目前欧姆龙主推以下 3 种类型变频器：3G3MZ-ZV2、3G3RX 和 3G3MX2。下面详细介绍欧姆龙最新发布和投入市场的 3G3MX2 变频器。

表 8.2 欧姆龙系列变频器特点比较

名 称	特 点
3G3JV 系列变频器	-支持开环 V/f 控制 -功率范围：0.1～3.7 kW -没有内置 RS422/485 通信口和 Modbus 协议 -面板上带有速度调节电位器，调速方便
3G3JZ 系列变频器	-支持开环 V/F 控制 -功率范围：0.2～3.7 kW -内置 RS422/485 通信口和 Modbus 协议 -面板上带有速度调节电位器，调速方便
3G3MZ-ZV2 系列变频器	-支持开环 V/F 控制和开环矢量控制 -功率范围：0.2～11 kW -内置 RS422/485 通信口和 Modbus 协议 -通过配置 DeviceNet 从站单元和 Profibus-DP 从站单元可实现 DeviceNet 和 Profibus 网络通信 -具有输入控制信号叠加功能、脉冲计数输出等功能 -面板上带有速度调节电位器，调速方便 -特殊功能：自动转矩补偿功能、频率叠加功能、PID 休眠功能、模拟量断线检测功能、加密功能、计数功能
3G3MV 系列变频器	-支持开环 V/F 控制和开环矢量控制 -功率范围：0.1～7.5 kW -内置 RS422/485 通信口和 Modbus 协议 -通过配置 DeviceNet 从站单元可以实现 DeviceNet 网络通信 -面板上带有速度调节电位器，调速方便
3G3RV-ZV1 系列变频器	-支持 V/F 控制和闭环矢量控制 -功率范围：0.4～300 kW -内置 RS422/485 通信口和 Modbus 协议 -通过配置 DeviceNet 从站单元和 Profibus-DP 从站单元可实现 DeviceNet 和 Profibus 网络通信 -具有转矩控制功能、零伺服功能、2 马达切换功能
3G3RX 系列变频器	-支持 V/F 控制和闭环矢量控制 -功率范围：0.4～400 kW -内置 RS485 通信口 -具有 0.3 Hz 时 200%以上的输出转矩 -带速度控制、转矩控制和位置控制 -带零伺服功能、张力控制、电子齿轮比设定、脉冲输入/输出、脉冲计数和简单逻辑编程功能
3G3MX2 系列变频器	-支持无传感器矢量控制和带速度反馈的 V/F 控制 -功率范围：0.1～15 kW -内置 RS485 通信口和 Modbus 协议 -具有 0.5 Hz 时 200%以上的输出转矩 -输出频率在高频模式下最大支持 1000 Hz -带简易定位功能、AVR 功能、V/f 特性切换、上下限限制、16 段多段速等功能

8.2.2 3G3MX2 系列变频器

3G3MX2 系列变频器是欧姆龙公司于 2009 年年末推出的紧凑型高功能小型变频器。

1. 3G3MX2 变频器的型号和种类

型号标准：

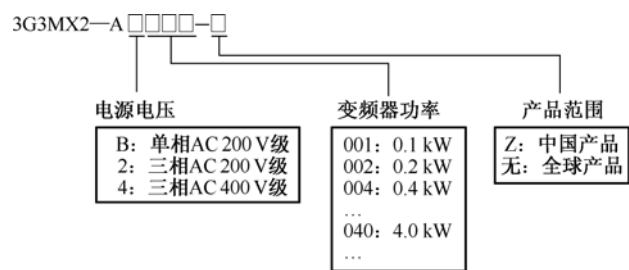


表 8.3 是 3G3MX2 系列变频器的全部种类。

表 8.3 3G3MX2 系列变频器

额 定 电 压	防 护 等 级	最大适用电动机容量		型 号	
		CT: 重载	VT: 轻载	全球产品	中国产品
三相 AC 200 V	IP20	0.1 kW	0.2 kW	3G3MX2-A2001	—
		0.2 kW	0.4 kW	3G3MX2-A2002	—
		0.4 kW	0.75 kW	3G3MX2-A2004	—
		0.75 kW	1.1 kW	3G3MX2-A2007	—
		1.5 kW	2.2 kW	3G3MX2-A2015	—
		2.2 kW	3.0 kW	3G3MX2-A2022	—
		3.7 kW	5.5 kW	3G3MX2-A2037	—
		5.5 kW	7.5 kW	3G3MX2-A2055	—
		7.5 kW	11 kW	3G3MX2-A2075	—
		11 kW	15 kW	3G3MX2-A2110	—
		15 kW	18.5 kW	3G3MX2-A2150	—
三相 AC 400 V	IP20	0.4 kW	0.75 kW	3G3MX2-A4004	3G3MX2-A4004-Z
		0.75 kW	1.5 kW	3G3MX2-A4007	3G3MX2-A4007-Z
		1.5 kW	2.2 kW	3G3MX2-A4015	3G3MX2-A4015-Z
		2.2 kW	3.0 kW	3G3MX2-A4022	3G3MX2-A4022-Z
		3.0 kW	4.0 kW	3G3MX2-A4030	—
		4.0 kW	5.5 kW	3G3MX2-A4040	3G3MX2-A4040-Z
		5.5 kW	7.5 kW	3G3MX2-A4055	3G3MX2-A4055-Z
		7.5 kW	11 kW	3G3MX2-A4075	3G3MX2-A4075-Z
		11 kW	15 kW	3G3MX2-A4110	3G3MX2-A4110-Z
		15 kW	18.5 kW	3G3MX2-A4150	3G3MX2-A4150-Z
单相 AC 200 V	IP20	0.1 kW	0.2 kW	3G3MX2-AB001	—
		0.2 kW	0.4 kW	3G3MX2-AB002	3G3MX2-AB002-Z
		0.4 kW	0.55 kW	3G3MX2-AB004	3G3MX2-AB004-Z
		0.75 kW	1.1 kW	3G3MX2-AB007	3G3MX2-AB007-Z
		1.5 kW	2.2 kW	3G3MX2-AB015	3G3MX2-AB015-Z
		2.2 kW	3.0 kW	3G3MX2-AB022	3G3MX2-AB022-Z

2. 3G3MX2 变频器的性能指标

不同型号的 3G3MX2 变频器的性能指标也不同，随着变频器功率增加，变频器重量、外形尺寸，以及变频器额定输出电流和额定输出容量都在增大。本书以 3G3MX2-A4040-Z 为例，叙述 3G3MX2 的操作、运行和通信，同样适合其他型号的 3G3MX2 变频器。

3G3MX2-A4040-Z 变频器的性能指标见表 8.4。

表 8.4 3G3MX2-A4040-Z 变频器的性能指标

适用 电机容量	额定 输出容量	额定 输入电压	额定 输出电压	额定 输出电流	短时间减速 时制动转矩 (未连放电电阻)	可连接最小 制动电阻值	尺寸、重量 (宽×高×长)
4.0 kW	6.0 kVA	三相 380×85% V~ 480×110% V 50/60×(1±5%) Hz	三相 380~480 V (不能超过 输入电压)	9.2 A	20%	100 Ω	140×128×170.5 mm 2.1 kg

3. 3G3MX2 变频器的特点

- 功率相同情况下，3G3JX、3G3RX 和 3G3MX2 这三种变频器相比较，3G3MX2 的外形尺寸最小，与 3G3RX 相比，宽度和高度尺寸缩减约一半。
- 容量大、系列齐全。3G3MX2 功率范围为 0.1~15 kW，JZ 和 MZ 系列变频器功率范围为 0.2~11 kW，RX 系列变频器功率范围为 0.4~400 kW。
- 高启动转矩：无传感器矢量控制，0.5 Hz 时启动转矩可达到额定转矩的 200%。
- 带速度反馈的 V/F 控制。
- 高速变频器通信 (Modbus)：不需要通信选装件便可进行多址通信，可执行多轴同时运行指令。
- 输出频率：高频模式下最大支持 1000 Hz。
- 简易定位功能：可实现 90°相位差编码器输入时的定位 (~32 kHz) 和单相脉冲输入时的定位 (~32 kHz)。
- 3G3MX2 可以实现全面的机械控制功能，如位置、速度同步以及逻辑程序设计。
- 3G3MX2 支持广泛的工业通信方式，可通过 Modbus、DeviceNet、Profibus、EtherCAT、Mechatrolink-II 等进行无缝的数据交换。

4. 变频器面板结构

3G3MX2 变频器正面如图8.7所示，图8.7(b)为带盖板的正面图，图8.7(c)为打开盖板的正面图，包括上部分的操作面板和下部分的接线端子。变频器的运行可直接通过设定操作面板来实现，也可以用 PLC 与变频器连接，通过变频器输入端子或通信端口两种通信方式实现变频器的运行控制。下面介绍操作面板各部分的名称及其功能。

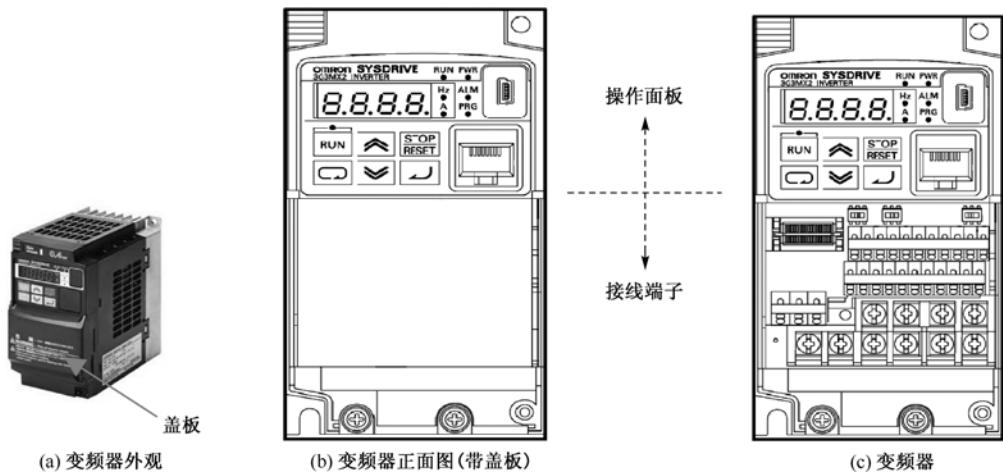


图 8.7 变频器面板

如图8.8所示为变频器的操作面板，操作面板是由数据显示、操作键盘、运行指示 LED、USB 连接器和 RJ45 连接器组成。

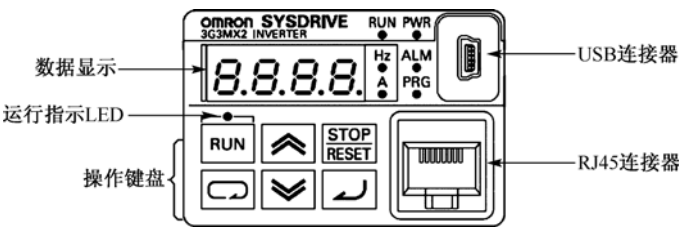


图 8.8 变频器操作面板

数据显示包括数码显示和简易指示灯。其中，数码显示部分主要用于显示：功能参数号、数据和故障代码，简易指示灯(PWR、ALM、PRG、RUN、Hz、A)主要用于显示变频器的状态。

操作面板各部分的名称和功能见表 8.5。

表 8.5 操作面板各部分名称和功能

操作器标记	名 称	功 能
PWR	电源指示灯	通电后灯亮 (绿色)
ALM	报警灯	变频器故障时灯亮 (红色)
PRG	编程指示灯	当数据显示部显示可变更数据时，灯亮 (绿色)；当设定值不当时，灯闪烁
RUN	运行指示灯	灯亮 (绿色) 时，表示变频器正在运行 (发出的命令在 0 Hz 下运行或者运行命令关闭后电动机减速运行，指示灯也会亮)
Hz	输出频率监视	灯亮 (绿色) 时，显示部显示频率 (Hz)
A	输出电流监视	灯亮 (绿色) 时，数码显示部分显示电流 (A)
8.8.8.8.	数码显示	显示频率和各种参数、设定值等数据
—●—	运行命令启用指示灯	灯亮 (绿色) 时，由操作面板控制变频器运行
RUN	运行键	启动变频器。此键只有在“由操作面板控制变频器运行”模式下有效
STOP RESET	停止/复位键	使变频器减速、停止。当 b087 设定为 01 (Disabled) 时，此键不起作用。在变频器发生异常时，它用于解除异常并且复位
↶	模式键	显示参数时：按下此键移到下一功能组的开头 显示数据时：按下此键取消设置，返回参数显示 独立输入模式时：按下此键向左移动闪烁位 不论当前显示什么，按住此键 1 s 以上，屏幕显示输出频率监视器 (d001) 的数据
↷ ↸	增/减键	按键，参数和设定数据增加/减少。长按可加速。同时按下两键，激活独立输入模式，在此模式，可通过模式键选择任意数字位进行编辑
↵	回车键 (输入确认键)	显示参数时，按下此键转到数据显示 显示数据时，确定/存储当前设置到 EEPROM 中，然后跳回参数显示 独立输入模式时：按此键，闪烁位右移
USB	USB 连接器	是计算机连接用 mini-B 型 USB 连接器。使用计算机通过 USB 连接器控制变频器时，可同时使用操作面板控制变频器运行
RJ45	RJ45 连接器	是选装的远程操作器用连接器 (RS-422)。连接远程操作器后，本体的按键即失效。此时，通过 b150 设定数据显示部显示的项目

5. 3G3MX2 接线端子和切换开关

变频器的接线端子和切换开关均位于操作面板盖的下方，如图8.9所示。打开变频器的面板盖，可见盖内的组成为：主电路端子块、控制电路端子块 A、控制电路端子块 B、继电器输出端子块、选装电路板安装连接器、终端电阻切换开关、安全功能切换开关、EDM 功能切换开关和充电 LED，其功能见表 8.6。

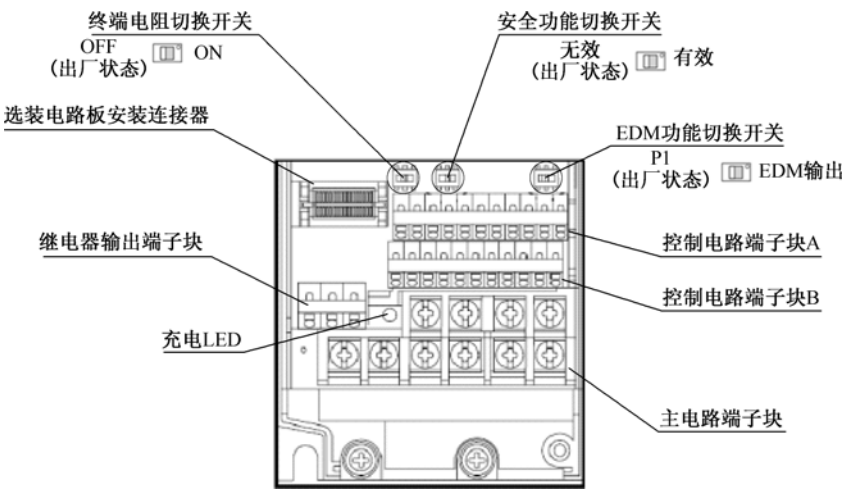


图 8.9 变频器的端子块

表 8.6 变频器的接线端子名称及说明 (功能)

名 称	说 明
终端电阻切换开关	是控制电路端子块 RS-485 端子的终端电阻切换开关。拨到“ON”位置后与内置的 200 Ω 电阻连接
安全功能切换开关	请在使用安全功能时拨到“ON”位置。请务必切断电源后再将开关拨到“ON”或“OFF”位置
EDM 功能切换开关	在使用安全功能的 EDM 输出时拨到“ON”位置。务必切断电源后再将开关拨到“ON”或“OFF”位置
选装电路板安装连接器	为选装电路板安装用连接器
控制电路端子块 A、B	是控制变频器用的各种数字量/模拟量输入/输出信号等的连接用端子块
继电器输出端子块	是继电器输出的 1c 触点端子块
主电路端子块	用于连接变频器主电源、变频器输出到电动机的电源和制动电阻器等
充电 LED	即使在电源断开后，主回路直流电压 (P/+2~N/-端子间) 达到约 45 V 以上时，灯也会亮。请在确认充电 LED 熄灭后再进行布线等作业

下面介绍各接线端子块。

(1) 主电路端子块

如图8.10所示为主电路端子块，各端子的说明见表 8.7。

(2) 控制电路端子块 A 和端子块 B

图8.11为 3G3MX2 控制电路端子块 A 和端子块 B，各端子的说明见表 8.8。

(3) 变频器标准连线

变频器标准连线如图 8.12 所示。

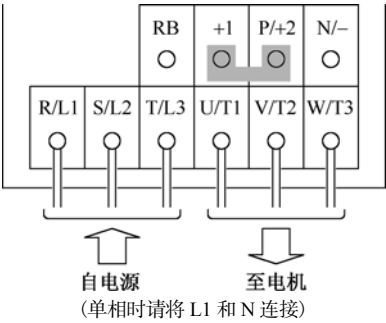


图 8.10 主电路端子块

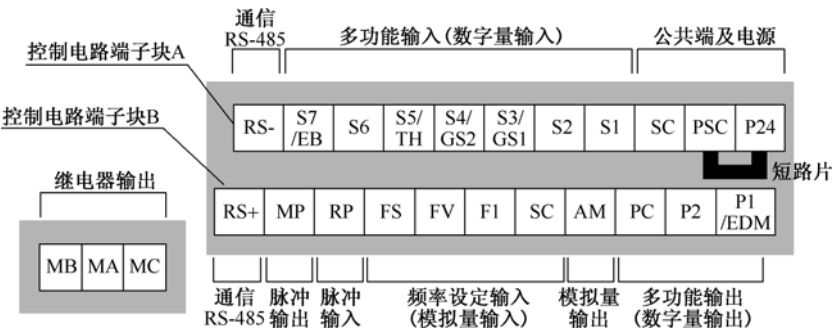


图 8.11 控制电路端子块 A 和端子块 B

表 8.7 主电路端子块的端子说明


端子记号	端子名称	内容说明
R/L1	主电源输入端子	连接输入交流电源 400 V 级 (当为单相 200 V 级型号时, 请连接至 L1 和 N)
S/L2		
T/L3		
U/T1	变频器输出端子	连接三相电动机
V/T2		
W/T3		
+1 (PD/+1)	DC 电抗器连接端子	拆下端子+1~P/+2 间的短路片, 连接选装的 DC 电抗器
P/+2 (P/+)		
P/+2 (P/+)	制动电阻器连接端子	连接选装的制动电阻器 (需要制动转矩时)
RB		
P/+2 (P/+)	再生制动单元连接端子	连接选装的再生制动单元 (需要制动转矩时或内置的制动电路不足时)
N/-		
	接地端子 (未画在图 8.10 中)	200 V 级采用 D 类接地, 400 V 级采用 C 类接地 200 V 级 3.7 kW 以下和 400 V 级 4.0 kW 以下的机型的接地端子位于散热片中

表 8.8 控制电路端子块的端子说明

		端子记号	端子名称	内容说明	规格
模 拟 量	电 源	SC (L)	输入信号用公共端	是内部电源、数字量输入、模拟量输入/输出端子的公共端子	—
		FS (H)	频率指令用电源	是 FV 端子用 DC 10 V 电源	允许最大电流 7 mA
	频率设定 输入 (模拟量 输入)	FV (O)	频率设定输入 (模拟量电压输入)	在 DC 0~10 V 输入电压下, 发送频率指令时使用	输入阻抗约 10 kΩ, 允许输入电压范围 DC -0.3~+12 V
		FI (OI)	频率设定输入 (模拟量电流输入)	在 4~20 mA 输入电流下, 发送频率指令时使用	输入阻抗约 100 Ω, 允许输入电流范围 0~24 mA
	传感器 输入	S5/TH (S/PTC)	外部热敏电阻输入 (与多功能输入端子兼用)	与 SC 间连接外部热敏电阻, 当温度呈异常状态时, 使变频器跳闸。 (热敏电阻约 3 kΩ 以上时跳闸)。因为兼用, 需要通过 C005 进行设定	PTC 类型
	模拟量 输出	AM	多功能模拟量输出 (电压)	可通过 DC 0~10 V 电压信号输出指定的信号	AM



(续表)

			端 子 记 号	端 子 名 称	内 容 说 明	规 格			
数 字 量	电 源		SC (L)	输入信号用公共端	是内部电源、数字量输入、模拟量输入/输出端子的公共端子	—			
			P24	输入信号用电源端子	是触点输入信号用 DC 24 V 电源。输入源型逻辑时为公共端	允许最大电流 100 mA			
			PSC (PLC)	输入端子用电源端子	输入漏型逻辑时，与 P24 短接；输入源型逻辑时，与 SC 短接。使用外部电源驱动触点输入时，请拆下短路片	—			
	数 字 量 输 入		触 点	S7/EB (7/EB) S6 (6) S5/TH (5/PTC) S4/GS2 (4/GS2) S3/GS1 (3/GS1) S2 (2) S1 (1)	多功能输入端子	选择 7 功能，分配给 S1～S7 端子。漏型逻辑、源型逻辑均支持	各输入与 PSC 间电压： ON 电压，最小 18 V OFF 电压，最大 3 V 允许最大电压，DC27 V 负载电流，5 mA (24 V 时)		
				S4/GS2 (4/GS2) S3/GS1 (3/GS1)	安全输入	安全功能切换开关 ON 时有效。详情请参见操作手册			
			脉 冲	RP (EA)	脉冲输入 A	是频率设定用脉冲输入 (内部电路与输入端子 S7/EB 不同，请予以注意)	输入脉冲 最高 32 kHz 输入与 SC 间电压： ON，最小 4 V OFF，最大 1 V 允许最大电压，DC 27 V		
				S7/EB (7/EB)	脉冲输入 B	是频率设定用脉冲输入 (内部电路与输入端子 RP 不同，请予以注意)	输入脉冲最高 1.8 kHz 输入与 SC 间电压： ON，最小 18 V OFF，最大 3 V 允许最大电压，DC 27 V 负载电流，5 mA		
			输 出		集 电 极 开 路	P1/EDM (11/EDM) P2 (12)	多功能输出端子	选择 2 功能，分配给 P1～P2 端子。漏型逻辑、源型逻辑均支持	集电极开路输出 各端子与 PC 间允许： 最大电压 27 V，最大电流 50 mA ON 时电压降 4 V 以下
						P1/EDM (11/EDM)	安全监控	EDM 功能切换开关 ON 时有效	
	继 电 器	MA MB			继电器输出端子	是 c (选择) 触点输出 继电器输出 (MA,MB) 触点选择的出厂设定值为 C036 = 01： 选 MA-MC 即为 b (常闭) 触点，选 MB-MC 即为 a (常开) 触点	触点最大容量 MA-MC: AC 250 V, 2 A (电阻), 0.2 A (感应) MB-MC: AC 250 V, 1 A (电阻), 0.2 A (感应) 触点最小容量 AC 100 V, 10 mA; DC 5 V, 100 mA		
		MC			继电器输出公共端				
	脉 冲	MP (EO)			脉冲输出	输出脉冲	输出脉冲最高 32 kHz，输出电压 DC 10 V 允许最大电流 2 mA		
串行通信			RS+ (SP) RS— (SN)	Modbus 端口 (RS-485)	RS-485 端口 RS+： RS-485 差动 (+) 信号 RS—： RS-485 差动 (—) 信号	最高速度 115.2 Kb/s, 内置终端电阻 200 Ω 滑动开关切换			

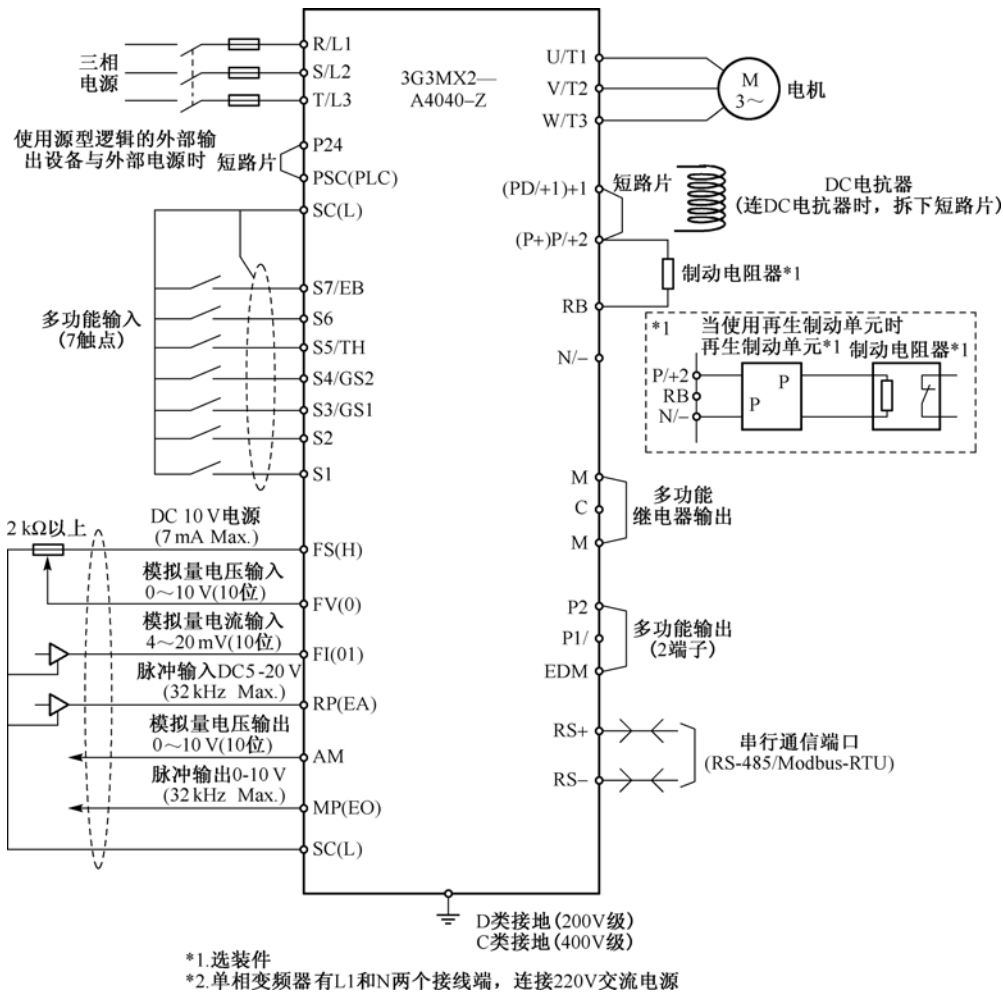


图 8.12 变频器标准连线图



8.3 3G3MX2 变频器的运行控制

3G3MX2 变频器的运行方式有操作面板控制、外部电路控制、PLC 直接控制和 PLC 通信控制。

8.3.1 变频器操作面板的操作

变频器共有 8 种参数功能组，每一功能组包括不同的参数号，如监控 d 参数功能组 d001.....d104、F 参数功能组 F001.....F004、A 参数功能组 A001.....A165、b 参数功能组 b001.....、C 参数功能组 C001.....、H 参数功能组 H001.....、P 参数功能组 P001.....、U 参数功能组 U001.....。变频器操作面板的操作分为基本操作和独立模式操作两种情况。

1. 基本操作

如图 8.13 所示。电源 ON 时，出厂设定为显示 d001(当前输出频率值)的数据。  
按键作用：  
按模式键移动到下一个参数功能组，或从数据返回到参数号但不保存数据。在独立输入模式下，按模式键，左移闪烁位。



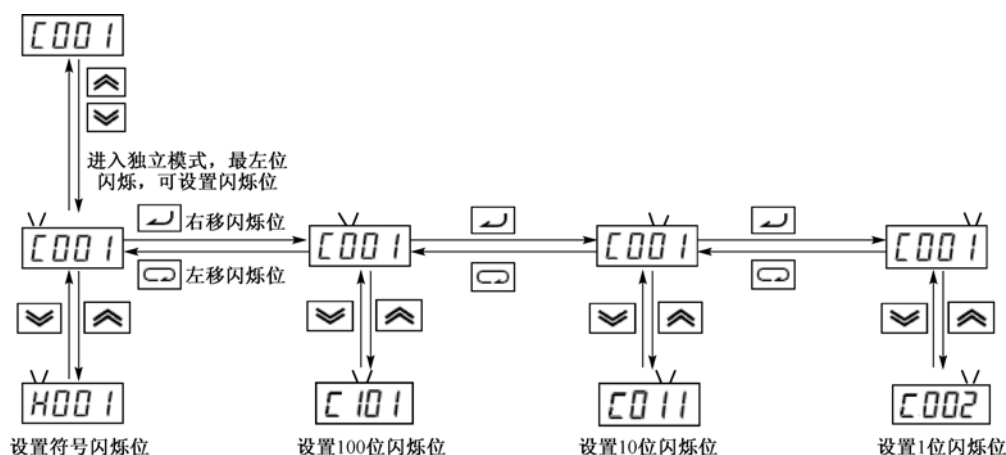


图 8.14 变频器面板操作的独立模式 (参数设置)

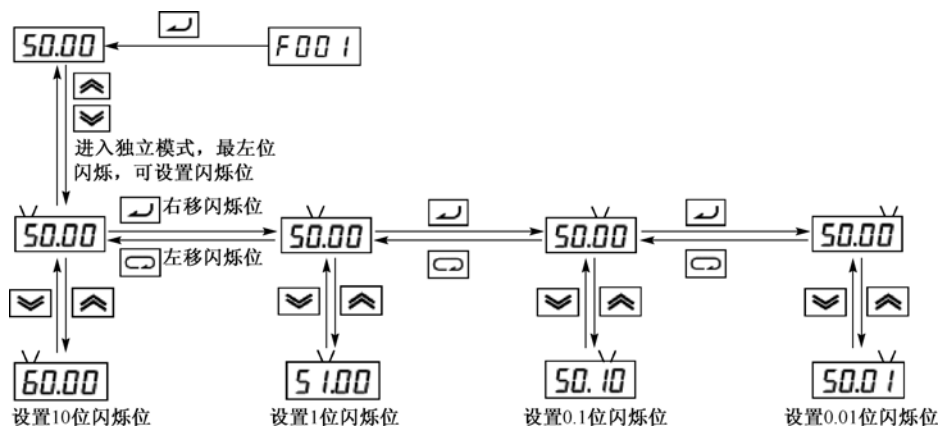


图 8.15 变频器面板操作的独立模式 (数据设置)

8.3.2 3G3MX2 运行的操作面板控制

3G3MX2 变频器的运行条件有两个：发出运行命令和进行频率设定。如发出运行命令而频率设定在 0 Hz，或者完成频率设定而没有发出运行命令，变频器不会工作。

通过操作面板的按键操作来控制变频器运行，设置见表 8.9。

表 8.9 操作面板按键操作变频器所需设置

参 数 号	功 能 名 称	数 据	默 认 值	说 明
A001	频率设定值选择 1	02 (用操作面板)	02	
A002	运行命令选择 1	02 (用操作面板)	02	运行命令启用时,指示灯 LED 亮
F001	输出频率设定	0.0 到最大频率	6.00	—
F004	运行方向选择	00 (正转) 01 (反转)	00	变频器运行时,此参数不能改变
F002	加速时间设定 1	0.01 到 99.99s 100.0 到 999.9s 1000.到 3600.s	10.00s	—
F003	减速时间设定 1	0.01 到 99.99s 100.0 到 999.9s 1000.到 3600.s	10.00s	—

**例 8-3-1** 如图 8.16 所示通过操作面板控制变频器运行接线示意图, 要求用操作面板控制变频器实现: ①电动机启、停和转速调节; ②电动机正转和反转控制。

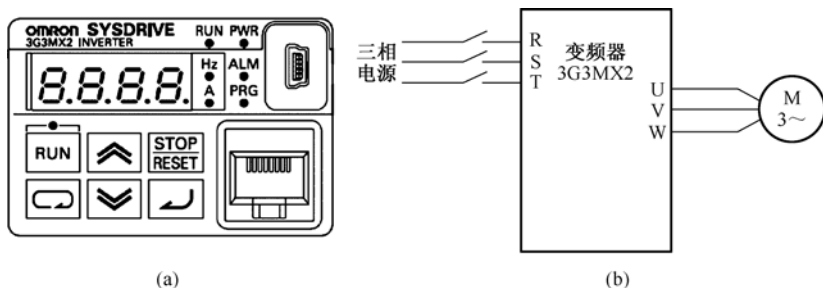


图 8.16 变频器操作面板和变频器接线示意图

步骤:

- ① 按图 8.15(b)正确接线后, 变频器加电, 此时变频器面板上显示 d001 的数据 0.00。
- ② 参数初始化, 设置 A001=02, A002=02, F001=2.5 Hz。
- ③ 设定运行方向选择 F004=00, 即设为正转。
- ④ 通过参数查找, 显示输出频率监控 d001。然后按回车键, 此时显示的是 0.00 Hz。或通过按模式键 1 s 以上, 就会显示 d001 的数据。
- ⑤ 按下运行键。此时运行指示灯变亮, 同时电动机开始正转。d001 显示的数据由 0.00 Hz 逐渐增大到 2.50 Hz。
- ⑥ 在电动机运行的状态下, 可以修改运行频率。找到参数 F001, 设置 F001=5 Hz, 立即观察到电动机转动加快。实现①。
- ⑦ 改变电动机的运行方向, 须停止电动机。按下停止 STOP/RESET 键, 电动机减速直至停止。找到参数 F004, 设置 F004=01, 即设电动机为反转。
- ⑧ 再按下运行键, 则电动机反转。实现②。

### 8.3.3 3G3MX2 运行的外部电路控制

通过将外部信号(开关、频率设定电位器等)连接到变频器控制电路接线端子, 控制变频器运行和频率设定。

#### 1. 2 线输入控制变频器运行

2 线输入控制是指采用两个多功能输入端(两根线)控制变频器的启、停和正反转, 并用模拟量电压/电流信号设定频率。用参数 A001、A002 和 A005 设定用控制电路接线端子控制变频器频率设定和运行。用多功能输入 S1、S2 做变频器(电动机)的正反运行控制开关, 须设置相应的 C001=00 和 C002=01; 选 S5 作为电压/电流频率设定选择, 则须设置相应的 A005=00 和 C005=16, 见表 8.10。当 A005=00 和 C005=16 时, S005 为 OFF 时为电压输入, 为 ON 时为电流输入。将电位器连接到模拟量输入端子 FV 和 FI 设定频率值, 外部控制电路端子接线如图 8.17 所示。

**注意:** 设置 C003~C007 中任意一个为值 16(FV/FI 输入), 如 C005, 表示其对应的输入端子 S5 设定为模拟电压/电流输入(FV/FI 输入)的选择开关, 用来选择用电压还是电流作为频率的设定值。选择电压(或电流)作为频率的设定值, 就要将另一个电流(或电压)模拟量输入端子与 SC 短接。当 A005=00 和 C005=16 时, 相应的 S005 为 OFF 时为电压输入, 为 ON 时为电流输入。如果设置区 C003~C007 都不为 16(FV/FI 输入), 频率设定值就是电压输入(FV)和电流输入(FI)的之和。

表 8.10 外部电路控制变频器所需设置

参 数 号	功 能 名 称	数 据	默 认 值	说 明
A001	频率设定值选择 1	01(用控制电路接线端子)	02	—
A002	运行命令选择 1	01(用控制电路接线端子)	02	—
A005	电流电压选择 (FV/FI 选择)	00(电压 FV/电流切换设定频率值)	00	—
C001	选择多功能输入 1(S1)	00(正转 FW: Forward)	00(FW)	输入端子改变, 则设定参数号变化, 例如输入端选 Sx, 则设定参数号为 C00x (x≠1)
C002	选择多功能输入 2(S2)	01(反转 RV: Reverse)	01(RV)	输入端子改变, 则设定参数号变化, 例如输入端选 Sx, 则设定参数号为 C00x (x≠2)
C005	选择多功能输入 5(S5) (作为 FV/FI 切换开关)	16(电压/电流 FV/FI 切换)	02	输入端子改变, 则设定参数号变化, 例如输入端选 Sx, 则设定参数号为 C00x (x≠5)
F002	加速时间设定 1	0.01 到 99.99s 100.0 到 999.9s 1000 到 3600.s	10.00s	—
F003	减速时间设定 1	0.01 到 99.99s 100.0 到 999.9s 1000 到 3600.s	10.00s	—

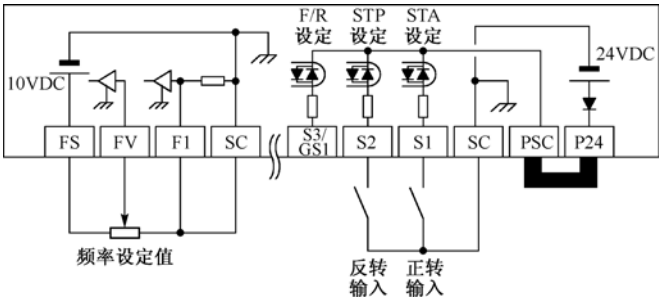


图 8.17 外部控制电路端子接线图(电压设定频率值)

例 8-3-2 如图8.18(a)和图8.18(b)所示接线图,用外部开关信号(2 输入端)按图 8.19 所示控制端 S1 和 S2 的动作时序,控制变频器的启停、加减速和正反转。画出运行指示和输出频率波形。

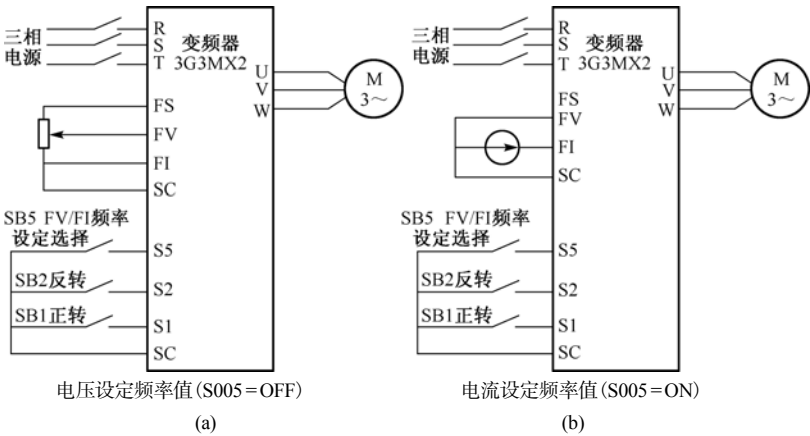


图 8.18 外部控制电路端子接线图

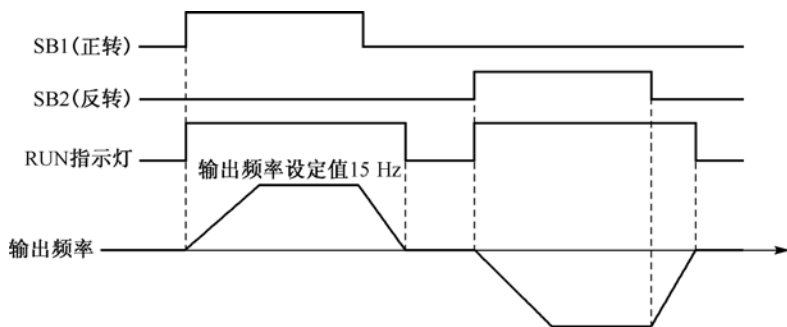


图 8.19 控制波形图

步骤：

- ① 按图8.18(a)或图8.18(b)接线，并接通变频器的电源。
  - ② 设置频率设定值选择 A001=01，运行命令选择 A002=01，电压/电流频率设定选择 A005=00。表明用变频器的控制电路接线端子控制其频率设定和运行。
  - ③ 用 2 个多功能输入 S1、S2 分别做变频器(电动机)的正反运行开关，设置相应的 C001=00(正转)和 C002=01(反转)。
  - ④ 选 S5 作为电压/电流频率设定选择，须设置相应的 C005=16。
  - ⑤ 显示输出频率监控(d001)然后按回车键。确定此时显示的是 0.00 Hz。
  - ⑥ 采用调节外接电阻电位器模拟量(电压/电流)设定频率，通过 F001 观察使频率设定值为 15.00 Hz。设置电动机的最大频率 A004=50 Hz，加速时间 F002=30 s，减速时间 F003=20 s。
- 按图 8.19 所示正转控制开关 SB1 和反转控制开关 SB2 的动作时序：
- ⑦ 合上正转控制开关 SB1=ON。此时运行指示灯亮，电动机用 9 s 加速到 15 Hz 的输出频率。
  - ⑧ 停留一段时间 10 s 后，断开正转控制开关 SB1=OFF，电动机用 6 s 减速到输出频率 0 Hz。停止后运行指示灯会熄灭。
  - ⑨ 停留一段时间 7 s 后，合上反转控制开关 SB2=ON。此时运行指示灯亮，电动机用 9 s 加速到输出频率 15 Hz。
  - ⑩ 停留一段时间例如 10 s 后，断开反转控制开关 SB2=OFF，电动机用 6 s 减速到 0 Hz。停止后运行指示灯会熄灭。

完成上述步骤即会得到图 8.19 所示的输出频率波形。

注意：所设的加减速时间 F002/F003 是从 0 Hz 到最高频率的加/减速时间。

合上正转控制开关 SB1=ON，待电动机加速到 15 Hz 的输出频率，逐步增大模拟量(电压/电流)设定的频率值，观察电动机的转速变化。再让 SB1=OFF，电动机减速直至停止。

本例用电压(电流)进行频率设定时，调节外接电阻电位器(或电流)，通过观察 F001 使频率设定为 15.00 Hz。下面介绍通过计算方法求频率设定值所对应的电压/电流设定值。

FV 电压/FI 电流与频率的对应：

通过计算方法使用模拟量电压输入 FV 或电流输入 FI 设定输出频率值时，要设置 A013 和 A014 的 FV 比例，或设置 A103 和 A104 的 FI 比例，以及设置 A004 中的最高频率值(对应 10 V 或 20 mA)，见表 8.11，如图 8.20 所示。图 8.20 的 A061 和 A062 分别是上下限频率。

表 8.11 使用模拟量电压输入 FV 或电流输入 FI 频率设置说明

参 数 号	A013	A014	A103	A104
功能名称	FV 起始比例	FV 终止比例	FI 起始比例	FI 终止比例
比率设定 (%)	0 (%)	100 (%)	20 (%)	100 (%)
默认值	0 (%)	100 (%)	20 (%)	100 (%)
对应电压/电流	0 V	10 V	4 mA	20 mA
对应频率	0 Hz	最高频率	0 Hz	最高频率
设定电压值/电流值	最高频率由 A004 设定，频率设定值 $f$ 与电压值和电流值的关系为 $U = \frac{10}{f_{A004}} f$ ， $I = \frac{20-4}{f_{A004}} f + 4$ (或 $f = \frac{f_{A004}}{10} U$ ， $f = \frac{I-4}{20-4} f_{A004}$ )			

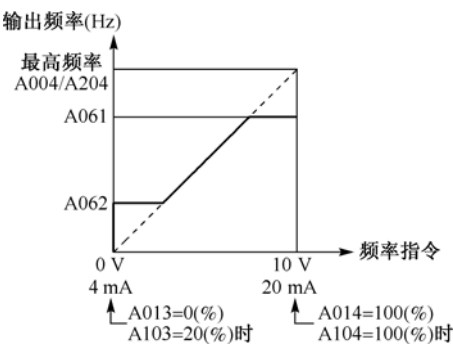


图 8.20 模拟量电压 FV/电流 FI 输入频率设置图

例 8-3-3 ①最高 A004=50 Hz 时，要设置输出频率 15 Hz，问电压/电流设置频率时，对应的电压/电流值应该为多少？②同样的电压/电流，在最高 A004 = 100 Hz 时，输出频率设定值分别为多少？

① A004=50 Hz，输出频率 15 Hz 时，对应电压值

$$U = \frac{10}{f_{A004}} f = \frac{10}{50} \times 15 = 3 \text{ V}$$

对应电流值

$$I = \frac{20-4}{f_{A004}} f + 4 = \frac{20-4}{50} \times 15 + 4 = 8.8 \text{ mA}$$

② A004=100 Hz，电压为 3 V 时，对应频率设定值

$$f = \frac{U}{10} f_{A004} = \frac{3}{10} \times 100 = 30 \text{ Hz}$$

电流为 8.8 mA 时，对应频率设定值

$$f = \frac{8.8-4}{20-4} \times 100 = 30 \text{ Hz}$$

2. 变频器运行的 3 线输入控制

3 线输入控制是指采用 3 个多功能输入端(3 根线)控制变频器的启、停和正反转，并用模拟量电压/电流信号设定频率。适用于变频器用按钮开关控制启动停止的情况。用参数 A001、A002 和 A005 设定用控制电路接线端子控制变频器频率设定和运行。用多功能输入 S1、S2 和 S3 分别做 STA(启动)、STP(停止)和 F/R(正传/反转选择)的控制端，须设置相应的 C001=20、C002=21 和 C003=22，选 S5 作为电压/电流频率设定选择，则须设置相应的 A005=00 和 C005=16，见表 8.12。当 A005=00 和 C005=16 时，S005 为 OFF 时选择电压输入，为 ON 时选择电流输入。将电位器连接到模拟量输入端子 FV 和 FI 设定频率值，3 线输入接线方法如图 8.21 所示。

表 8.12 3 线输入控制变频器所需设置

参数号	功 能 名 称	数 据	默 认 值	说 明
A001	频率参考选择 1	01(控制电路接线端子)	02	—
A002	运行命令选择 1	01(控制电路接线端子)	02	—



(续表)

参数号	功 能 名 称	数 据	默 认 值	说 明
A005	电流电压选择 (FV/FI 选择)	00(电压 FV/电流切换设定频率值)	00	—
C001	多功能输入 1 选择 (S1)	20(启动 STA: 3 线启动)	00 (FW)	输入端子改变, 则设定参数号变化, 例如输入端选 Sx (x≠1), 则设定参数号为 C00x
C002	多功能输入 2 选择 (S2)	21(停止 STP: 3 线停止)	01 (RV)	输入端子改变, 则设定参数号变化, 例如输入端选 Sx (x≠2), 则设定参数号为 C00x
C003	多功能输入 3 选择 (S3)	22(正反转 F/R: 3 线正转/反转)	18 (RS)	输入端子改变, 则设定参数号变化, 例如输入端选 Sx (x≠3), 则设定参数号为 C00x
C005	选择多功能输入 5 (S5)	16(FV/FI: 切换)	02	输入端子改变, 则设定参数号变化, 例如输入端选 Sx (x≠5), 则设定参数号为 C00x

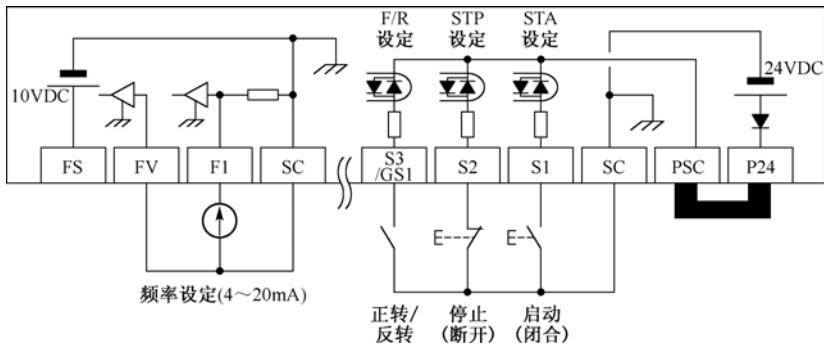


图 8.21 3 线输入接线图

注意: 设置 C003~C007 中任意一个为值 16(FV/FI 输入), 如 C005, 表示其对应的输入端子 S5 设定为模拟电压/电流输入 (FV/FI 输入) 的选择开关, 用来选择用电压还是电流作为频率的设定值。选择电压(或电流)作为频率的设定值, 就要将另一个电流(或电压)模拟量输入端子与 SC 短接。当 A005=00 和 C005=16 时, 相应的 S005 为 OFF 时为电压输入, 为 ON 时为电流输入。如果设置区 C003~C007 都不为 16(FV/FI 输入), 频率设定值就是电压输入 (FV) 和电流输入 (FI) 之和。

例 8-3-4 如图 8.22 所示接线, 用外部开关信号 (3 输入端) 按图 8.23 所示控制端 S1、S2 和 S3 的动作时序, 控制变频器的启停、加减速和正反转。画出输出频率波形。

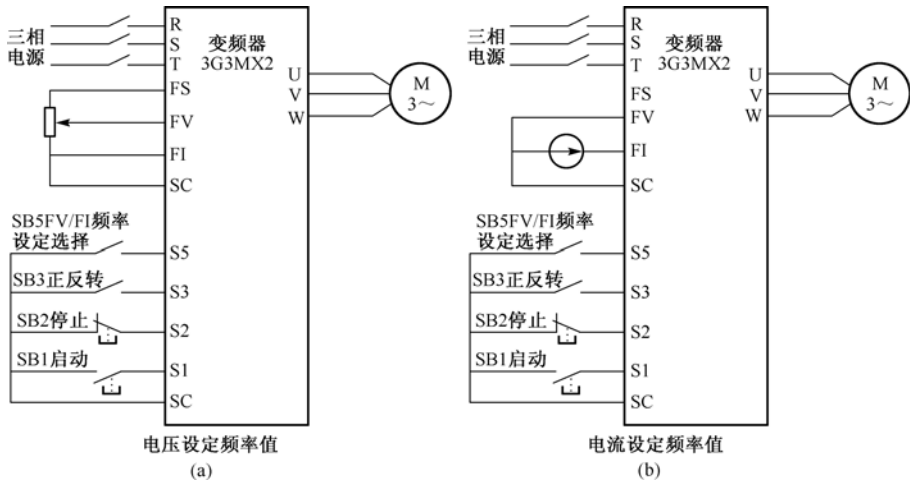


图 8.22 外部控制电路端子接线图

- ① 按图 8.22 接线，并接通变频器的电源。
- ② 设置频率设定值选择 A001=01，运行命令选择 A002=01，电压/电流频率设定选择 A005=00。表明用变频器的控制电路接线端子控制其频率设定和运行。
- ③ 用 3 个多功能输入 S1、S2 和 S3 分别做 STA(启动)、STP(停止)和 F/R(正传/反转选择)的控制端，须设置相应的 C001=20(STA)、C002=21(STP)和 C003=22(F/R)。
- ④ 选 S5 作为电压/电流频率设定选择，设置相应的 C005 = 16。当 S005 为 OFF 时为电压输入，为 ON 时为电流输入。
- ⑤ 显示输出频率监控(d001)然后按回车键。确定此时显示的是 0.00 Hz。
- ⑥ 按公式  $U = \frac{10}{f_{A004}} f$  或  $I = \frac{20-4}{f_{A004}} f + 4$  通过电位器设定频率的模拟量(电压/电流)值，使设定频率为 15.00 Hz。设置电动机的最大频率 A004=50 Hz，加速时间 F002=30 s，减速时间 F003=20 s。

按图 8.23 所示，SB1、SB2 和 SB3 的动作时序：

- ⑦ 合上停止开关 SB2=ON 做启动准备，让启动开关 SB1 ON 一个脉冲。此时电动机逐渐加速到 15 Hz，并维持此速度正转运行。
- ⑧ 停留一段时间 10 s 后，接通正反转切换开关 SB3=ON，电动机逐渐减速到 0 Hz，继续反转到 15 Hz，并维持此速度反转运行。
- ⑨ 停留一段时间例如 10 s 后，断开停止开关，SB2 OFF 一个脉冲。电动机逐渐减速到 0 Hz。完成图 8.23 操作过程。

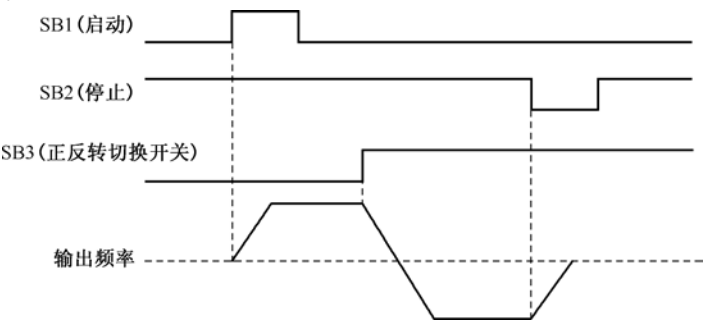


图 8.23 3 线输入控制过程

合上停止开关 SB2=ON 做启动准备，正转控制开关 SB1=ON，待电动机加速到 15 Hz 的输出频率，逐步减小模拟量(电压/电流)设定的频率值，观察电动机的转速变化。再让 SB2=OFF，电动机减速直至停止。

3. 与运行有关的其他功能

- (1) 2 段加减速功能(2CH)  
通过此功能旨在加减速过程中改变加减速时间，见表 8.13。加减速时间的切换方法有以下 3 种。
  - ① 通过多功能输入端子切换。  
若需要通过多功能输入端子进行加减速切换，请将 C003~C007 中任意一个置为 09(2CH)，例如，C005 置为 09，表示其对应的多功能输入端子 S5 为加减速切换端子。
  - ② 通过任意频率自动切换。
  - ③ 只在正反转切换时自动切换。

表 8.13 2 段加减速功能设置

参数 No.	功 能 名 称	数 据	默 认 值	单 位
A092/A292	第 1/ 第 2 加速时间 2	0.01~3600.(例 8-3-1、例 8-3-2)	10.00	s
A093/A293	第 1/ 第 2 减速时间 2	0.01~3600.(例 8-3-1、例 8-3-2)	10.00	s
A094/A294	2 段加减速选择	00:通过 2CH 端子进行切换(例 8-3-5)	00	—
		01:通过 2 段加减速频率进行切换(例 8-3-6)		
		02:仅正反转切换时有效(例 8-3-7)		
A095/A295	2 段加速频率	0.00~400.0(1000.) 2 段加减速选择(A094/A294) 设为 01 时有效(例 8-3-6)	0.00	Hz
A096/A296	2 段减速频率	0.00~400.0(1000.) 2 段加减速选择(A094/A294) 设为 01 时有效(例 8-3-6)	0.00	Hz
C001~C007	多功能输入功能选择	09: 2CH (2 段加减速)	—	—
相关功能	—	F002/F202、F003/F203	—	—

例 8-3-5 通过输入端子切换 (A094/A294=00), C005=09, 则对应 S5 端为加减速控制端, 如图 8.24 所示。

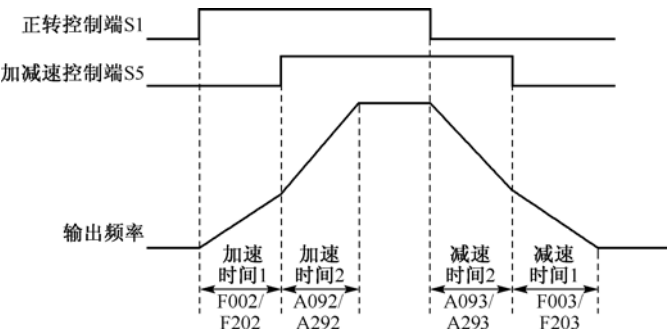


图 8.24 输入端子切换加减速

例 8-3-6 通过频率切换 (A094/A294=01), 当输出频率达到 A095/A096 中设定值时, 改变加速/减速时间, 如图 8.25 所示。

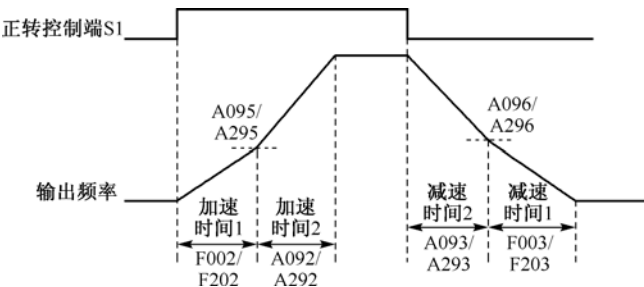


图 8.25 通过频率切换切换加减速

例 8-3-7 通过正反转切换 (A094/A294=02), 如图 8.26 所示。

(2) 加减速模式  
本功能通过 A097、A098 选择模式, 可以设定加速时或减速时的加减速模式。加减速模式选择直

线(A097/A098 = 00)以外时,如果在模拟量输入(A001 = 01)下使用本功能,加减速时间会延长,请不要使用,见表 8.14 和表 8.15。

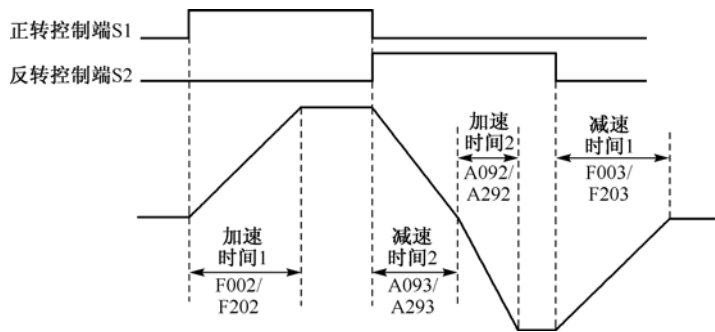


图 8.26 通过正反转切换切换加减速

表 8.14 加减速模式设置

参数 No.	功 能 名 称	数 据	默 认 值	单 位
A097/A098	加速模式选择/ 减速模式选择	00: 直线	01	—
		01: S 形曲线		
		02: U 形曲线		
		03: 反 U 形曲线		
		04: EL-S 形曲线		
A131/A132	加速曲线常数/ 减速曲线常数	01 (弯曲程度小) ~ 10 (弯曲程度大) EL-S 形 (A097/A098 = 04) 以外时有效	02	—
A150/A151	EL-S 形加速时 曲线比率 1/2	0 ~ 50 指定使用 EL-S 形时曲线部分的比率(加速时用)	10	%
A152/A153	EL-S 形减速时 曲线比率 1/2	0 ~ 50 指定使用 EL-S 形时曲线部分的比率(减速时用)	10	%

表 8.15 加减速模式说明

设 定 值	00	01	02	03	04
曲线	直线	S 形	U 形	反 U 形	EL-S 形
A097 (加速)					
A098 (减速)					
内容	以直线加减速, 直至达到输出频率设定值	有助于防止升降机、传送带等的货物倒塌	有助于卷绕机等张力控制、防止卷绕物断裂		与 S 形一样, 可实现无振动开始、停止, 中间为直线

8.3.4 CP1H 通过输出端子直接控制变频器

尽管可以直接通过开关和模拟量输入控制变频器。在多数场合, 变频器通常与上位机如 PLC、PT 或 PC 等设备构成控制系统, 以实现复杂的控制功能。变频器的频率设置虽然可以从变频器的模

拟输入端子送入，进行无级调速。但在实际自动控制系统中，其频率设定信号、运行控制等信号往往来自 PLC。

PLC 对变频器的控制一般有两种方式，第一种是用变频器自带端子控制。PLC 的输出端子接变频器的多功能端子，变频器中设置多功能端子为多段速功能，并设置相应频率。通过 PLC 的输出端子的闭合和断开的组合，使变频器在不同速度下运行。这种控制方式的优点是响应速度快，抗干扰能力强。缺点是不能无级调速。第二种是将 PLC 的通信接口与变频器的通信接口相连，PLC 作为主站使用 Modbus-RTU 简易主站可以简单地通过串行通信来控制变频器等支持 Modbus 的从站设备。这需要设置 PLC 的通信协议和变频器的通信协议、通信地址、传输速率和奇偶码校验等。这种控制方式的优点是：可以无级变速，速度变换平滑，速度控制精确，适应能力好。缺点是抗干扰差，响应有延时。

8.3.3 节中介绍了通过外部开关控制变频器电路接线端子，从而控制电动机运行的方法。实际上外部开关可以由 PLC 的输出点代替，由 PLC 输出点的通断代替开关的闭合和断开，实现对变频器及电动机的控制。这就是 PLC 通过输出端子控制变频器。

本节介绍 PLC 通过输出端子对变频器进行多段速选择控制，并举例说明。多段速选择控制功能是向指定的参数区设定多个运行速度，通过多功能输入端子的通断状态组合来选择速度。多段速运行可以通过设置 A019 选择 4 个多功能输入端子( $S_{i+3}$ 、 $S_{i+2}$ 、 $S_{i+1}$  和  $S_i$ )控制的最大 16 段速运行或 7 个多功能输入端子( $S_7 \sim S_1$ )最大 8 段速运行，见表 8.16 所示。

表 8.16 多功能输入端设定多段速运行说明

参 数 号	功 能 名 称	数 值	默 认 值
A019	多段速选择	00: 通过 4 个端子进行 16 段速的选择	00
		01: 通过 7 个端子进行 8 段速的选择	
A020	第 1 多段速 0 速设定值	0.00、启动频率~最高频率	6.00 Hz
A021~A035	1~15 段速设定值	0.00、启动频率~最高频率	0.00 Hz
A220	第 2 多段速 0 速设定值	0.00、启动频率~最高频率	6.00 Hz
C001~C007	多功能输入功能选择	02~05: 二进制运行 16 速(CF1~CF4)	—
		32~38: 位运行 8 速(SF1 ~ SF7)	—
C169	多段速·位置	0~200.( $\times 10$ ms)	0
	确定时间	到端子输入确定为止的等待时间	

16 段速：见表 8.16，通过让对应的参数区  $C_{i+3}=05$ (CF4)、 $C_{i+2}=04$ (CF3)、 $C_{i+1}=03$ (CF2) 和  $C_i=02$ (CF1) ( $C_7 \sim C_1$  中任意相连的 4 个)，选择多功能输入端  $S_{i+3}$ 、 $S_{i+2}$ 、 $S_{i+1}$  和  $S_i$  4 个端子(对应  $S_7 \sim S_1$  中相连的 4 个)控制 16 段速 0~15 速。4 个多功能输入端设定 16 段速的逻辑见表 8.17。通过参数区 A020~A035 设定 0~15 速的频率，见表 8.16。多段速运行优先于频率设定值选择(A001)。但 0 速的频率则遵守频率设定值选择(A001)的设定内容。

8 段速：见表 8.16，通过让对应的参数区  $C_7 \sim C_1=38 \sim 32$ (SF7~SF1)，选择多功能输入端  $S_7 \sim S_1$  7 个端子控制多段速 0~7 速。7 个多功能输入端设定 8 段速的逻辑见表 8.18。要通过参数区 A020~A027 设定 0~7 速的频率，见表 8.16。多段速运行优先于频率设定值选择(A001)。但 0 速的频率则遵守频率设定值选择(A001)的设定内容。

变频器的多段速选择功能，能够控制电动机的连续运行和断续运行，还可以对各种速度任意设置加减速时间、加减速模式和运行时间等。因此，这种功能经常应用到提升装置中。

表 8.17 4 个多功能输入端设定 16 段速逻辑表

速度号 \ 端子号	多功能输入端子(S <sub>7</sub> ~S <sub>4</sub> )中任意 4 个, 如 S <sub>6</sub> ~S <sub>3</sub>			
	S <sub>6</sub> (CF4)	S <sub>5</sub> (CF3)	S <sub>4</sub> (CF2)	S <sub>3</sub> (CF1)
多段速 0	OFF	OFF	OFF	OFF
多段速 1	OFF	OFF	OFF	ON
多段速 2	OFF	OFF	ON	OFF
多段速 3	OFF	OFF	ON	ON
多段速 4	OFF	ON	OFF	OFF
多段速 5	OFF	ON	OFF	ON
多段速 6	OFF	ON	ON	OFF
多段速 7	OFF	ON	ON	ON
多段速 8	ON	OFF	OFF	OFF
多段速 9	ON	OFF	OFF	ON
多段速 10	ON	OFF	ON	OFF
多段速 11	ON	OFF	ON	ON
多段速 12	ON	ON	OFF	OFF
多段速 13	ON	ON	OFF	ON
多段速 14	ON	ON	ON	OFF
多段速 15	ON	ON	ON	ON

表 8.18 7 个多功能输入端设定 8 段速逻辑表

速度号 \ 端子号	多功能输入端子 S <sub>7</sub> ~S <sub>1</sub>						
	S <sub>7</sub> (SF7)	S <sub>6</sub> (SF6)	S <sub>5</sub> (SF5)	S <sub>4</sub> (SF4)	S <sub>3</sub> (SF3)	S <sub>2</sub> (SF2)	S <sub>1</sub> (SF1)
多段速 0	OFF	OFF	OFF	OFF	OFF	OFF	OFF
多段速 1	×	×	×	×	×	×	ON
多段速 2	×	×	×	×	×	ON	OFF
多段速 3	×	×	×	×	ON	OFF	OFF
多段速 4	×	×	×	ON	OFF	OFF	OFF
多段速 5	×	×	ON	OFF	OFF	OFF	OFF
多段速 6	×	ON	OFF	OFF	OFF	OFF	OFF
多段速 7	ON	OFF	OFF	OFF	OFF	OFF	OFF

例 8-3-8 提升系统。

在如图8.27所示的矿井提升系统的速度曲线中，它将提升运行分为启动、加速至稳定运行、减速至匀速运行和制动 4 个阶段。料车提升一次所需时间与料车的运动速度、加速度以及矿井深度有关。在不同阶段，它们的加速度大小由工艺要求确定，则运行时间根据行程距离来计算。

提升系统的运行速度曲线如图 8.28 所示。在第 1 阶段(启动)0~ $t_2$ ，A 料车从井底开始上升，同时 B 料车从井口位置开始下行。这个阶段的加速过程 1 取 10 s，使变频器输出达 25 Hz，电动机按中速 25 Hz 运行。在第 2 阶段(加速) $t_2 \sim t_4$ ，A 料车上行并加速到最大速度，加速过程 2 取 5 s，变频器输出最高频率 50 Hz，并稳定运行。在第 3(减速)阶段 $t_4 \sim t_6$ ，A 料车进入卸料前的第一次减速，减速过程 1 取 4 s，使变频器输出达 10 Hz，电动机按低速运行。在第 4(制动)阶段 $t_6 \sim t_7$ ，A 料车进入卸料前的第二次减速，直至停车，此时减速过程 2 取 5 s。

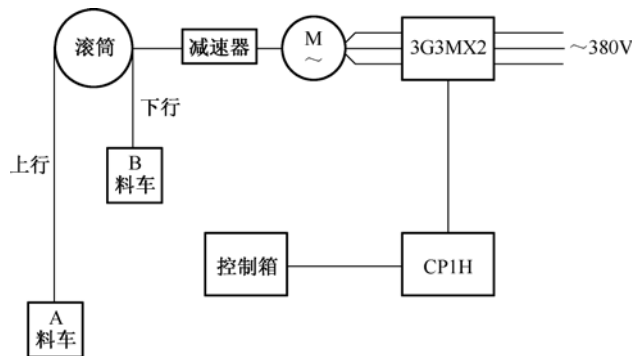


图 8.27 提升系统示意图

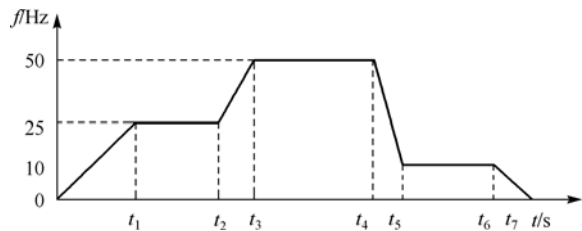


图 8.28 提升系统的运行速度曲线

- 根据上述系统的工作情况，归纳起来有 3 个基本要求：
- ① 用按钮通过 PLC 来控制电动机的正/反转，使货物上行和下行。
  - ② 能够改变变频器的加减速时间，以改变提升货物的加速度。
  - ③ 提供低速、中速和高速三种运行速度，供用户选择。

1. CPH 的 I/O 分配

CPH 的 I/O 分配见表 8.19。表中 SB1 为 A 料车上行按钮，SB2 为 B 料车上行按钮，SB3、SB4 为低、中、高速切换按钮，SB5 为加减速时间切换按钮。CPH 的输出点 100.01~100.05，直接与变频器的多功能输入端 S1~S5 连接，由 CPH 输出接点的状态来代替开关，实现对变频器和电动机的调速控制。

表 8.19 I/O 地址分配表

输入元件	功能说明	PLC 输入地址	变频器输入端子	功能说明	PLC 输出地址
SB1	A 料车上行	0.01	S1	正转/停止	100.01
SB2	B 料车上行	0.02	S2	反转/停止	100.02
SB3	速度切换输入 1	0.03	S3	多段速设定输入 1	100.03
SB4	速度切换输入 2	0.04	S4	多段速设定输入 2	100.04
SB5	加减速时间切换	0.05	S5	加减速时间切换控制	100.05
ER	MB 故障输入	0.00	—	—	—

2. 变频器与 PLC 的连接

根据 I/O 分配以及系统的控制要求，CPH 与 3G3MX2 的外部接线如图 8.29 所示。其中，3G3MX2 的运行状态由 CPH 控制，S1 控制电动机正转，S2 控制电动机反转，S3、S4 为变频器的 4 段速控制输入端（4 段速只需 2 个输入端控制），S5 选用加减速时间 2。另外，ER 为系统故障输入，由变频器的 MB、MC 控制。注意：所有输出点的 COM 口应与 SC 端子连接。

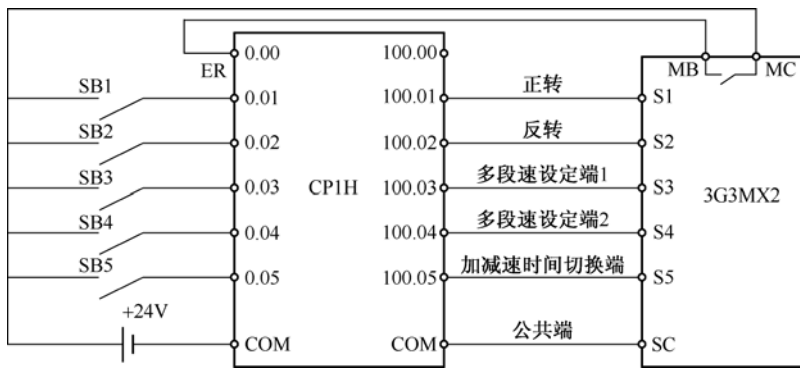


图 8.29 电路的外部接线

3. 参数设置

在本例中选择 A019=00：二进制选择(用 4 端子进行 16 段速选择)；由于只有 0.00 Hz、10.00 Hz、25.00 Hz 和 50.00 Hz 4 种频率，对应电动机的 4 个转速，所以实际只用 2 个控制端子 S3 和 S4 按 BCD 编码，就可以得到 4 种转速。

为便于程序设计和外部接线，本系统中采用 SB3、SB4 按钮通过 CP1H PLC 对变频器的 S3、S4 进行控制，以实现变频器对电动机的 4 段速调节，见表 8.20。

表 8.20 4 段速逻辑表

状态 端子	S4	S3	FOUT
0 段速	0	0	0 Hz
1 段速	0	1	25 Hz
2 段速	1	0	50 Hz
3 段速	1	1	10 Hz

变频器的参数设置如下：

- A004=50 Hz，最高频率；
- A061=50.00 Hz，频率上限 50 Hz，A062=0.00 Hz，频率下限 0 Hz；
- C036=01 (NC contact at P1, P2, MA, NO contact at MB)，通电时变频器异常闭合 MB-MC；
- A019=00，二进制选择(用 4 端子进行 16 段速选择)；
- A020=0.00 Hz，多段速 0 速；
- A021=25.00 Hz，多段速 1 速；
- A022=50.00 Hz，多段速 2 速；
- A023=10.00 Hz，多段速 3 速；
- F002=20.00 s，加速时间 1 为 20 s；
- A092=10.00 s，加速时间 2 为 10 s；
- F003=25.00 s，减速时间 1 为 25 s；
- A093=5.00 s，减速时间 2 为 5 s；
- A094=00，通过 2 段加减速端子切换；
- A001=01，A002=01，A005=00，通过外部端子电路设置运行频率和运行命令；
- C001=00 (FW)，S1 为正转控制端；
- C002=01 (RV)，S2 为反转控制端；



C003=02(CF1)，S3 为多段速设定端 1；  
C004=03(CF2)，S4 为多段速设定端 2；  
C005=09(2CH)，S5 为加减速时间切换端。

注意：所设的加减速时间 F002/F003 和 A092/A093 是从 0 Hz 到最高频率的加/减速时间。这里设最高频率 A004=50 Hz，则加速时第 1 段 0 Hz 至 25 Hz 需 10 s(S5=OFF)，第 2 段 25 Hz 至 50 Hz 需 5 s(S5=ON)，减速时 50 Hz 至 10 Hz 为 4 s(第 2 段，S5=ON)，10 Hz 至 0 Hz 为 5 s(第 1 段，S5=OFF)。

4. PLC 梯形图程序

(1) 手动程序

该系统通过 PLC 程序来控制变频器的多段速运行。实现图 8.28 的功能。手动 PLC 梯形图程序如图 8.30 所示。

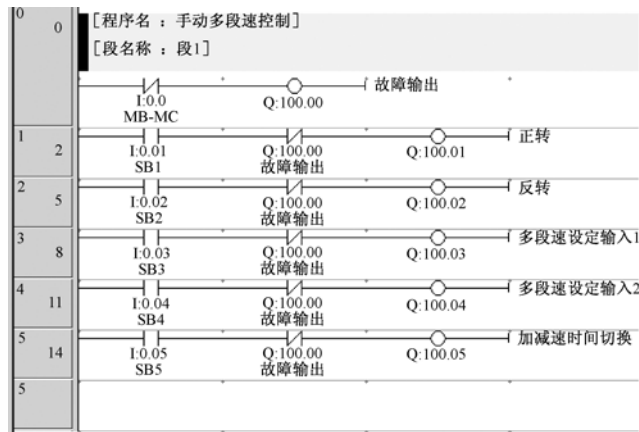


图 8.30 手动 PLC 梯形图程序

阶段 1：接通 0.01 和 0.03，断开 0.05。  
阶段 2：0.01 保持接通，同时断开 0.03，接通 0.04 和 0.05。  
阶段 3：0.01、0.04 和 0.05 保持接通，再接通 0.03。  
阶段 4：0.01 保持接通，断开 0.03、0.04 和 0.05。  
以上 4 个阶段构成图 8.28 中提升系统的运行速度曲线。

(2) 自动程序

如图 8.31 所示，采用自动程序时，图 8.29 中 SB3、SB4 和 SB5 失去作用。

5. 图 8.29 电路中按钮作用的解释

- ① 输入点 0.00 的状态由 MB、MC 控制，平时断开。当变频器出现故障时则 0.00 闭合，由 100.00 输出警报信号，并使其他输出接点都断开；直到故障消除，系统才恢复正常工作。
- ② 0.01(SB1)闭合，100.01 接通，则电动机正转使 A 料车上行；0.02(SB2)闭合，即 100.02 接通，电动机反转使 B 料车上行；0.03(SB3)和 0.04(SB4)组成多段速控制。当 SB4 断开、SB3 闭合时，系统以 25 Hz 中速运行；当 SB4 闭合、SB3 断开时，系统以 50 Hz 高速运行；当 SB4 和 SB3 均闭合时，系统以 10 Hz 低速运行；当 SB4 和 SB3 均断开时，系统减至 0 速。
- ③ 0.05(SB5)为加减速时间切换。当 SB5 断开时，100.05 接点断开，系统处于加/减速时间 1；当 SB5 闭合时，100.05 接点闭合，系统处于加/减速时间 2。

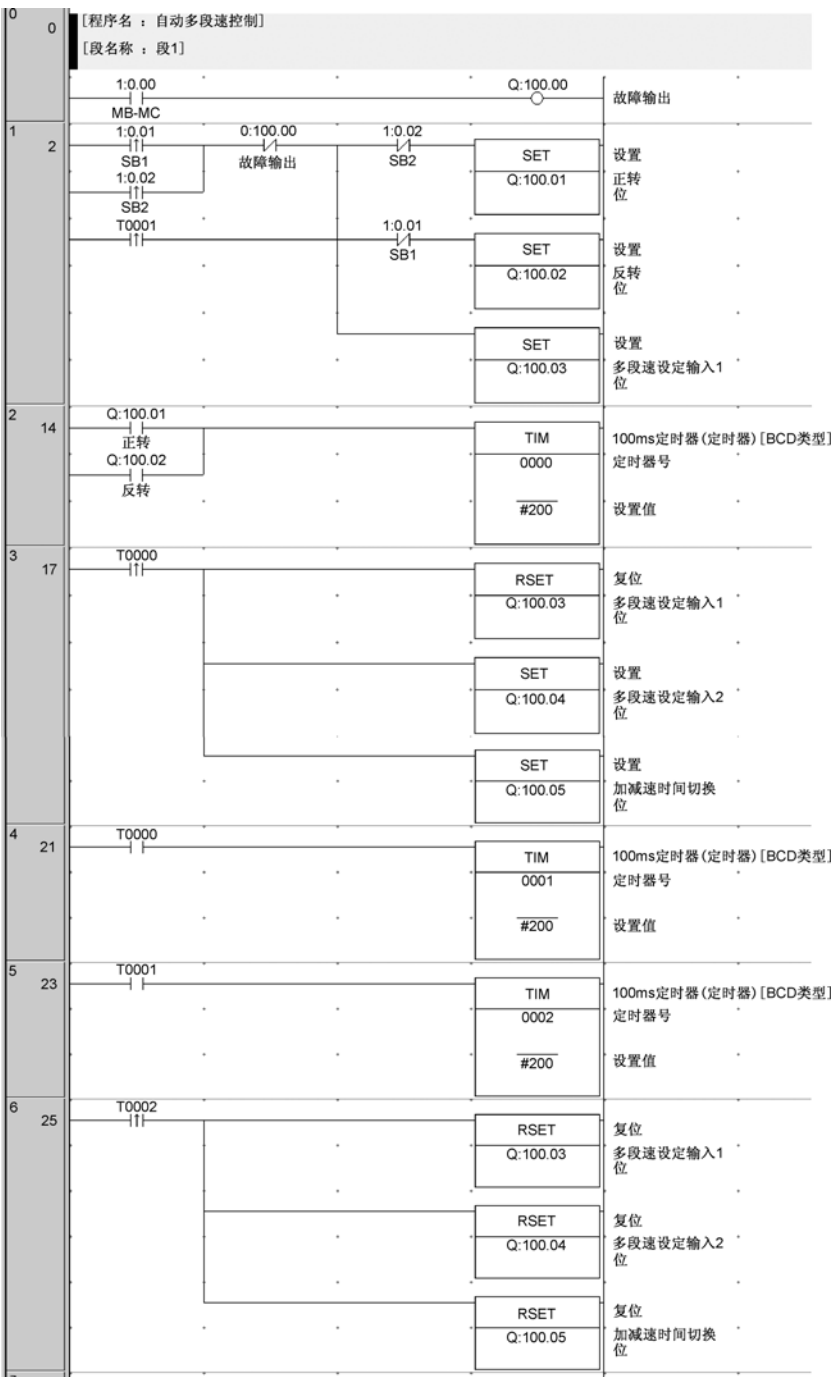


图 8.31 自动多段速控制

8.3.5 CP1H 通过 Modbus-RTU 简易主站功能控制变频器

Modbus 协议广泛应用于各种工业控制设备中，欧姆龙的变频器 3G3MX2、3G3JV、3G3MV 和 3G3RV 都支持 Modbus-RTU 通信，因此通过 CP1 系列的串行通信接口，可以实现 CP1 系列的 Mosbus-RTU 简易主站功能和支持 Mosbus-RTU 的变频器从站进行通信。

CP1H 的 CPU 单元上有两个串行端口:串口 1、串口 2,这两个串口都可选用端口插件 CP1W-CIF01 配置为 RS232C 口,或选用端口插件 CP1W-CIF11 配置为 RS422/485 口。CP1H 的 CPU 单元上配置的 RS422/485 口与 3G3MX2 变频器上的 RS422/485 口相连,采用国际标准的 Modbus 协议进行主从通信。通过编写 PLC 程序,使变频器作为 Modbus 协议从站接收来自 CP1H PLC 主站的通信指令,实现起停、频率给定、监控等功能。

CP1H 与 3G3MX2 系列变频器的通信需要做如下工作。

### 1. 硬件连接

确认 CP1H 已安装好 RS485 通信选件。CP1H 的 CPU 单元上有两个串行端口:串口 1、串口 2,这两个串口都可选用端口插件 CP1W-CIF01 配置为 RS232C 口,或 CP1W-CIF11 配置为 RS422/485 口。现选 CP1W-CIF11 配置为 RS485,并安装到通信端口 2 上。

如图 8.32 所示,CP1H 与一台变频器的连接,使用双绞屏蔽电缆连接 CP1W-CIF11 和 3G3MX2 变频器,电缆的一端接在 CP1H 插件 CIF11 的 SDA-、SDB+ 端子,另一端接 3G3MX2 变频器控制电路端子块的 RS-、RS+ 端子上,其余线屏蔽不用。

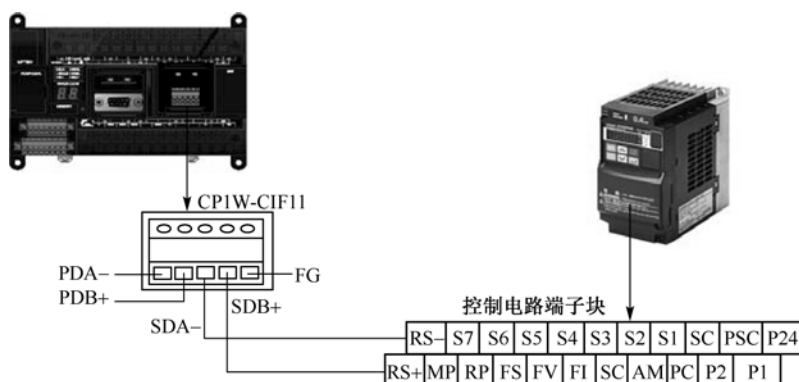


图 8.32 CP1H 与 1 台 3G3MX2 变频器的连接

图 8.33 为 CP1H 与多台变频器的连接。为使 RS485 通信保持稳定,将变频器终端电阻切换开关拨到 ON 的位置⑤

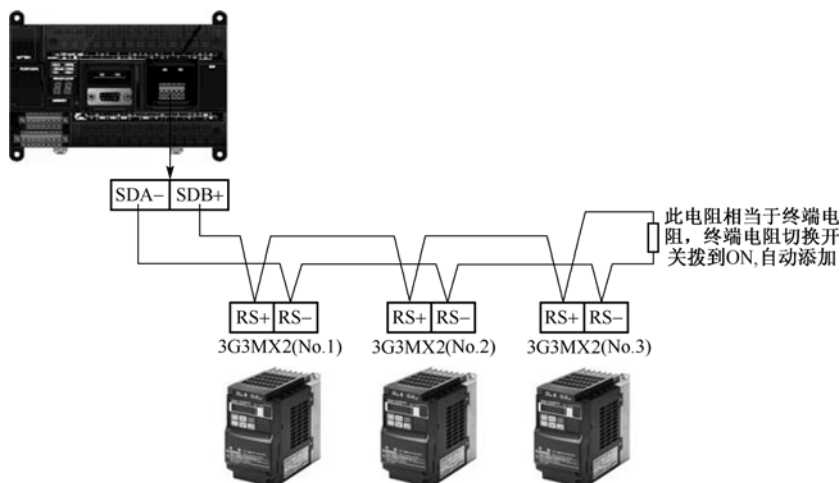


图 8.33 CP1H 与多台 3G3MX2 变频器的连接

2. CP1W-CIF11 设定

CP1W-CIF11 设定开关见表 8.21。

表 8.21 CP1W-CIF11 设定开关

开 关	含 义		说 明
SW1	ON	有	终端电阻有无的选择
	OFF	无	
SW2	ON	2 线(RS485)	SW2 和 SW3 设置方式相同
	OFF	4 线(RS422)	
SW3	ON	2 线(RS485)	
	OFF	4 线(RS422)	
SW4	—	—	空置
SW5	ON	RD: 有 RS 控制	选择 RD 的 RS 控制的有无
	OFF	RD: 无 RS 控制	
SW6	ON	RD: 有 RS 控制	1N 情况下, 四线式时, SW6 为 ON; 两线式时, SW6 为 ON
	OFF	SD: 无 RS 控制	

CP1H 使用 Modbus-RTU 简易主站功能时，CP1W-CIF11 开关设定如下。

SW1: ON，使用终端电阻。

SW2、SW3: ON，采用 RS485 方式。

SW4: OFF，空置。

SW5: 任意。

SW6: ON，发送有 RS 控制。

3. CP1H 设定

通信设置：以 PLC 通过串口 2 与变频器连接为例，使用编程软件 CX-Programmer 将串口 2 模式设置为“串口网关”，通信波特率为 9600，数据格式为 8，1，E，如图 8.34 所示。

参数区设置：须设置 Modbus-RTU 简易主站的参数区，见表 8.22。

Modbus-RTU 简易主站串口通信特殊辅助继电器说明见表 8.23。

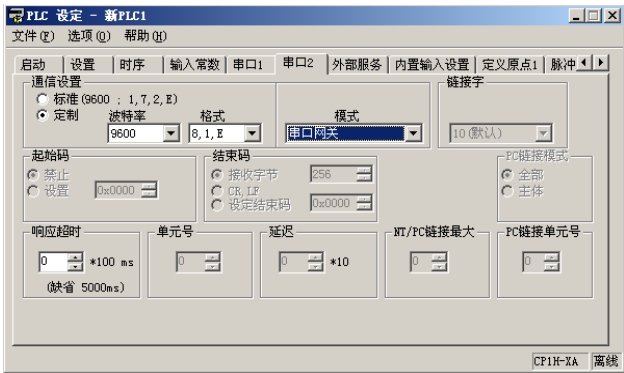


图 8.34 CP1H 串口 2 设置

串口 1 在 D32200~D32249 中保存要发送给变频器的 Modbus-RTU 命令。CX-P 与 CP1H 建立在线连接后，当串口 1 的通信使能位 A641.00 由 0→1 时，Modbus-RTU 命令自动发出，变频器返回的响应保存在 D32250~D32299 中；串口 2 在 D32300~D32349 中保存要发送给变频器的 Modbus-RTU 命令，CX-P 与 CP1H 建立在线连接后，当串口 2 的通信使能位 A640.00 由 OFF→ON 时，Modbus-RTU 命令自动发出，变频器返回的响应保存在 D32350~D32399 中。

表 8.22 Modbus-RTU 简易主站的参数区

通 道		位	设 定 内 容	
串 口 1	串 口 2			
D32200	D32300	07~00	命令	从站地址(00H~F7H)
		15~08		系统保留(固定设为 00H)
D32201	D32301	07~00		功能代码
		15~08		系统保留(固定设为 00H)
D32202	D32302	15~00		通信数据字节数(0000H~005EH)
D32203~D32249		15~00		通信数据(最大 94 字节)
D32250	D32350	07~00	响应	从站地址(01H~F7H)
		15~08		系统保留(固定设为 00H)
D32251	D32351	07~00		功能代码
		15~08		保留
D32252	D32352	07~00		出错代码
		15~08		系统保留(固定设为 00H)
D32253	D32353	15~00		应答字节数(0000H~03EA H)
D32254~D32299		15~00		应答(最大 92 字节)

表 8.23 Modbus-RTU 简易主站串口通信特殊辅助继电器

特殊辅助继电器通道		位	设 定 内 容
串口 1 对应	串口 2 对应		
A641CH	A640CH	00	Modbus-RTU 简易主站功能执行开关 0→1: 执行开始 1: 执行中 0: 非执行中或执行结束
		01	Modbus-RTU 简易主站功能执行正常结束标志 1: 执行正常结束 0: 执行异常结束或执行中
		02	Modbus-RTU 简易主站功能执行出错结束标志 1: 执行异常 0: 执行正常结束或执行中

4. 3G3MX2 设定

端电阻切换开关设定：将变频器终端电阻切换开关拨到 ON 的位置，目的是使 RS485 通信保持稳定，如图 8.35 所示。

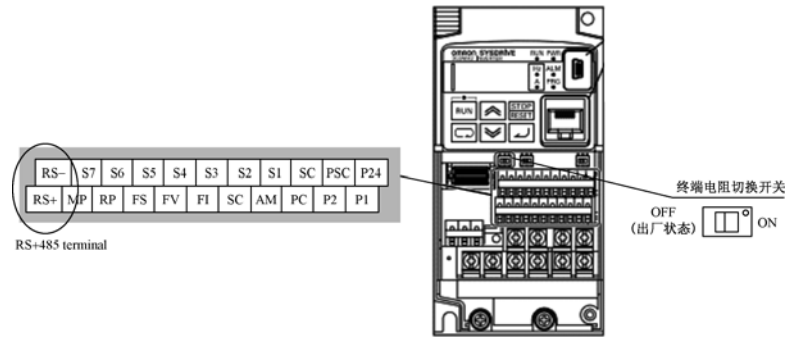


图 8.35 3G3MX2 变频器 RS485 通信设置

变频器通信参数设定如下。

A001: 03 (频率参考选择 1 选择 Modbus 通信)。

A002: 03 (运行命令选择 1 选择 Modbus 通信)。

C071: 05 (通信速率 9600 Kb/s)。

C072: 1. (通信站号选择 1)。

C074: 01 (通信奇偶校验选择偶校验)。

C075: 1 (通信停止位选择 1 位停止)。

C077: 0.00 (通信错误超时时间为 0.00 s)。

C078: 0. (通信等待时间为 0 ms)。

5. CP1H 使用 Modbus-RTU 简易主站功能与变频器通信

以串口 2 为例，介绍 CP1H 向变频器写通信参数、写频率、停止变频器运行和读取变频器运行频率。每一次通信都需要让使能位 A640.00 由 0→1，通信正常完成则会看到 A640.01 的值变为 1，通信过程执行了将保存在 D32300~D32349 中的 Modbus-RTU 命令发送给变频器，变频器返回的响应保存在 D32350~D32399 中。

(1) 写通信参数

在向变频器写入频率指令之前需要打开向位写入开关，其格式为在 D32300~D32304 分别填写数据 0001 (从站地址)、0005 (写入位操作)、0004 (通信数据字节4)、0000 (通信位地址=位编号 (0001H)-1)、#FF00 (OFF→ON，位 0001H 为 1，启动运行)，其中 0004 是标准 Modbus 协议内没有的数据，指 0000 和 FF00 共 4 字节。将 A640.00 置 1，会看到 A640.01 的值变为 1，表明执行正常，执行结果如图 8.36 所示。

地址	数据类型...	功...	值	
D32300	CHANNEL ...		0001 十六进制	命令数据
D32301	CHANNEL ...		0005 十六进制	
D32302	CHANNEL ...		0004 十六进制	
D32303	CHANNEL ...		0000 十六进制	
D32304	CHANNEL ...		FF00 十六进制	
D32350	CHANNEL ...		0001 十六进制	响应数据
D32351	CHANNEL ...		0005 十六进制	
D32352	CHANNEL ...		0000 十六进制	
D32353	CHANNEL ...		0004 十六进制	
D32354	CHANNEL ...		0000 十六进制	
D32355	CHANNEL ...		FF00 十六进制	
A640.00	BOOL (On...		0	
A640.01	BOOL (On...		1	← 使能位
A640.02	BOOL (On...		0	
A640.03	BOOL (On...		0	

图 8.36 写通信参数

(2) 写频率

在D32300~D32304分别填写如下数据：0001 (从站地址)、0006 (写入寄存器)、0004 (通信数据字节 4)、0001 (寄存器地址 = 寄存器编号 (0002H)-1)、01F4，意思为向 1 号变频器写入频率 5 Hz (1F4h=500 d=5.00 Hz)，其中 0004 是标准 Modbus 协议内没有的数据，指 0001 和 1F4 共 4 个字节，如图 8.37 所示。

(3) 停止变频器运行

停止变频器运行时，可以将变频器的频率设置为 0 Hz，或者单独向变频器发送停止命令。向变频器写入停止命令来停止运行，即向变频器寄存器的 0001H 写入 0000。具体说是在 D32300~D32304 分别填写如下数据：0001 (从站地址)、0005 (写入位操作)、0004 (通信数据字节 4)、0000 (通信位地址 = 位编号 (0001H)-1)、#0000 (ON→OFF，位 0001H 为 0，停止运行)。

(4) 读频率

如图 8.38 所示，从变频器读频率的高位和低位等，一共读取变频器两个寄存器：0001H(高位)为 0000 和 0002H(低位)为 01F4。其正确结果需要通过移位整理，要注意。

地址	数据类...	功...	值
D32300	CHANNEL ..		0001 十六进制
D32301	CHANNEL ..		0006 十六进制
D32302	CHANNEL ..		0004 十六进制
D32303	CHANNEL ..		0001 十六进制
D32304	CHANNEL ..		01F4 十六进制
D32350	CHANNEL ..		0001 十六进制
D32351	CHANNEL ..		0006 十六进制
D32352	CHANNEL ..		0000 十六进制
D32353	CHANNEL ..		0004 十六进制
D32354	CHANNEL ..		0001 十六进制
D32355	CHANNEL ..		01F4 十六进制
A640.00	BOOL (0...)		0
A640.01	BOOL (0...)		1
A640.02	BOOL (0...)		0
A640.03	BOOL (0...)		0

图 8.37 写频率

地址	数据类...	功...	值
D32300	CHANNEL ..		0001 十六进制
D32301	CHANNEL ..		0003 十六进制
D32302	CHANNEL ..		0004 十六进制
D32303	CHANNEL ..		0000 十六进制
D32304	CHANNEL ..		0002 十六进制
D32350	CHANNEL ..		0001 十六进制
D32351	CHANNEL ..		0003 十六进制
D32352	CHANNEL ..		0000 十六进制
D32353	CHANNEL ..		0005 十六进制
D32354	CHANNEL ..		0400 十六进制
D32355	CHANNEL ..		0001 十六进制
D32356	CHANNEL ..		F400 十六进制
A640.00	BOOL (0...)		0
A640.01	BOOL (0...)		1
A640.02	BOOL (0...)		0
A640.03	BOOL (0...)		0

图 8.38 读频率

8.3.6 利用 CX-Drive 软件控制变频器

1. 连线

CX-Drive 是欧姆龙 CX-One 软件包中独立的控制变频器和伺服驱动器的软件，CX-Drive 的具体功能有如下三点：

- ① 设定、下载、上载和对比参数。
- ② 测试运行和调整。
- ③ 对变频器和伺服驱动器进行监控和数据跟踪。

CX-Drive 1.7 以上版本支持 3G3MX2，本书使用了 CX-One4.0 中的 CX-Drive 1.92 版。3G3MX2 利用计算机 USB 口与安装于上位计算机的 CX-Drive 通信，如图 8.39 所示。

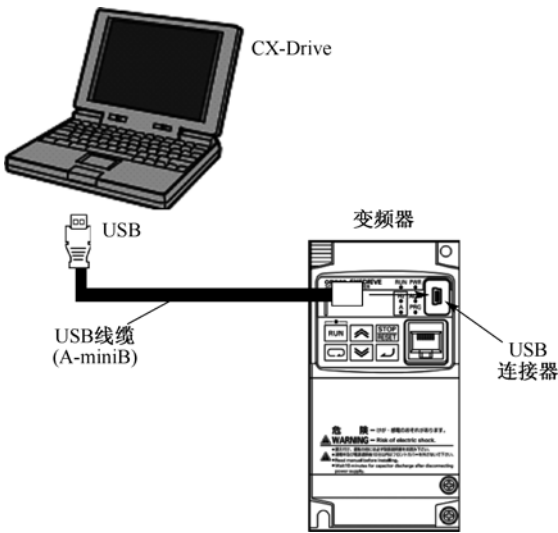


图 8.39 3G3MX2 通过计算机 USB 口和 CX-Drive 通信

2. 设置变频器型号

启动 CX-Drive，界面如图 8.40 所示。

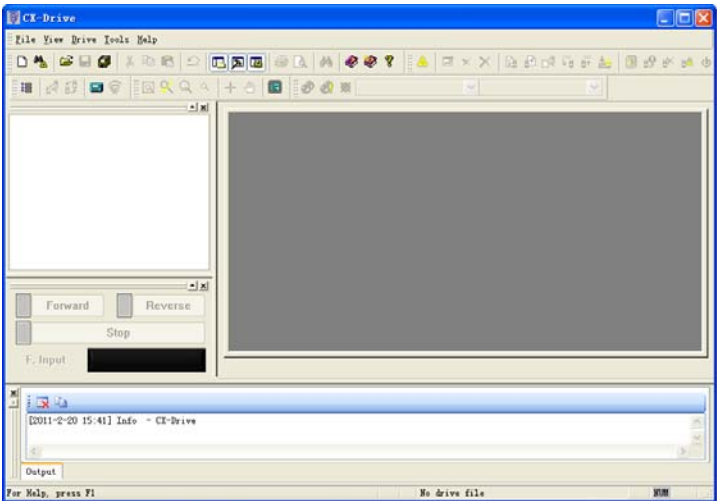



图 8.40 CX-Drive 启动界面

单击工具栏中的 New 按钮, 新建工程, 出现如图8.41所示的“New Drive”对话框, 对话框中有 4 个选项。在“Drive Name”文本框中输入名称, 在“Drive Type”选项中分别选择: “Inverter”和“3G3MX2”; 在“Connection Type”下拉列表中选择“Direct(USB)”, 如图 8.41 所示。

单击型号右边的“Settings”按钮, 出现“Drive Type Settings[3G3MX2]”(变频器类型设定)对话框, 如图 8.42 所示。在此对话框中有“Installation type/Option”(安装类型选择)、“Voltage Class”(单相或三相)、“Maximum Motor Capacity”(功率)、“Software Number”(软件版本)等选项。按实际使用的变频器规格选择后, 单击“OK”按钮, 得到如图8.43所示 3G3MX2 设定界面。界面中含有工作区、电动机管理窗口、输出窗口和编辑监视窗口。界面的右下角可以看到变频器的型号。

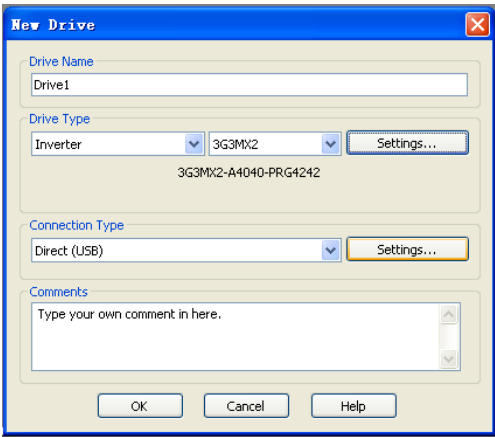


图 8.41 “New Drive”对话框

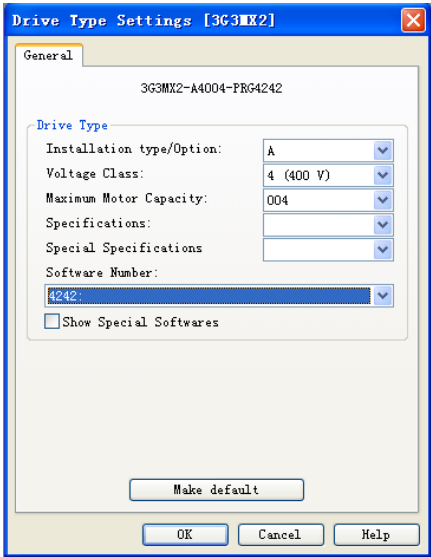


图 8.42 变频器类型设定对话框

或者选择自动检测模式, 在菜单栏 Drive 选项卡里选择 AutoDetection, 可以自动检测出连接的变频器型号, 如图 8.44 所示。



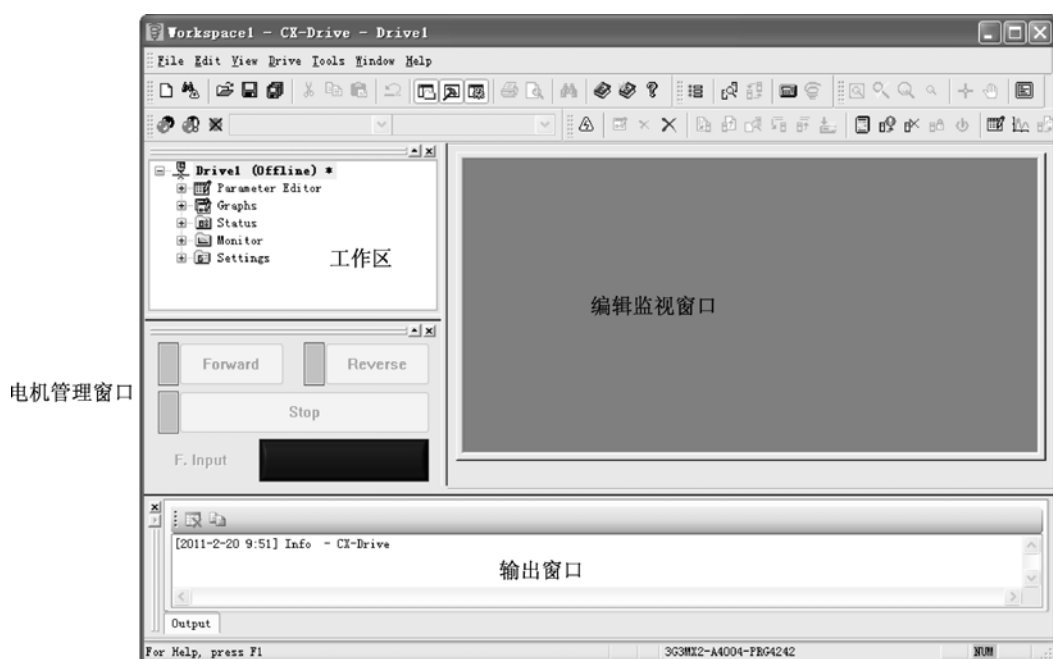


图 8.43 3G3MX2 设定界面

### 3. CX-Drive 功能简介

如图 8.43 所示，工作区中的项目以分层树形结构显示，分层树形结构可以通过 压缩或者扩展。工作区中树形根 Drive 的目录项目有 Parameter Editor、Graphs、Status、Monitor 和 Settings。自动检测模式窗口如图 8.44 所示。

选择“Parameter Editor”（参数编辑），在右边的编辑窗口中，可以对变频器的参数进行编辑，包括 A、b、C、d、F、H 和 P 等参数功能组的编辑。

选择“Graphs”（曲线）：在右边的编辑窗口中，可以得到 V/F 曲线。

选择“Status”（状态）：在右边的编辑窗口中，可以显示数字输入状态、数字输出状态、驱动状态和警报。

选择“Monitor”（监控）：在右边的编辑窗口中，可以进行实时监控。

选择“Setting”（设置）中的“Initialize”可以将全部参数初始化为默认值。

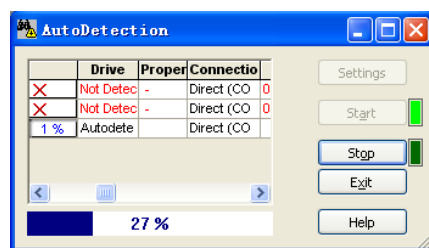


图 8.44 自动检测模式窗口

### 4. CX-Drive 的在线工作

单击工具栏中的 按钮，可以在线编辑变频器参数，如图 8.45 所示。

在线工作时，单击“Drive”→“Transfer”→“To Drive”或者“From Drive”命令，上载或下载变频器的参数，如图 8.46 所示。数据传送如图 8.47 所示。

CX-Drive 软件可以实现面板设置能实现的全部功能，如参数设置、多功能输入、模拟量输入/输出的监控。此外，还具有以图表或方框图形式显示参数、在线状态显示、监控、数据追踪、测试运行以及 FFT 分析等功能。

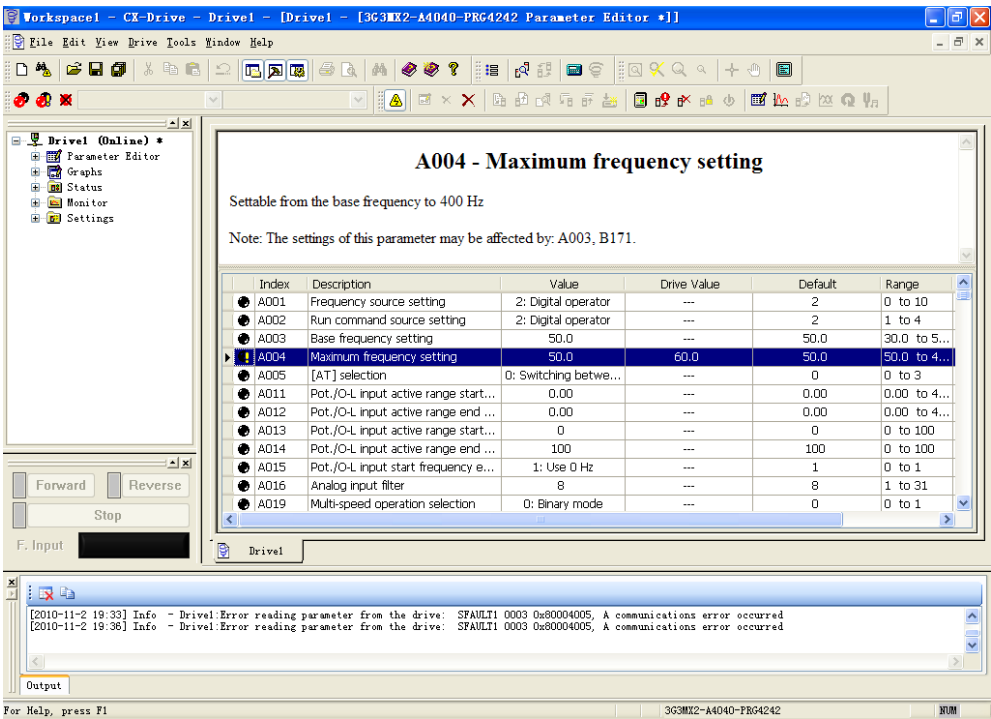


图 8.45 CX-Drive 在线工作模式

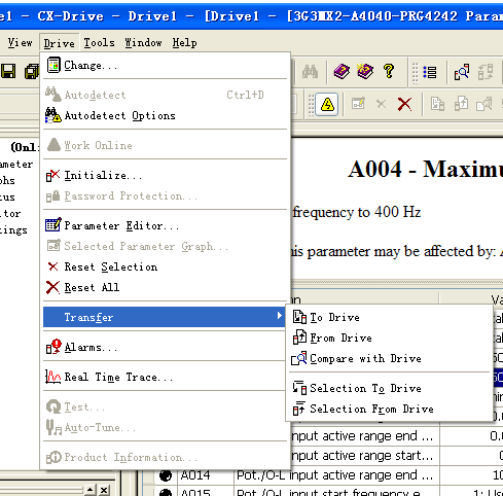


图 8.46 CX-Drive 的上载和下载操作

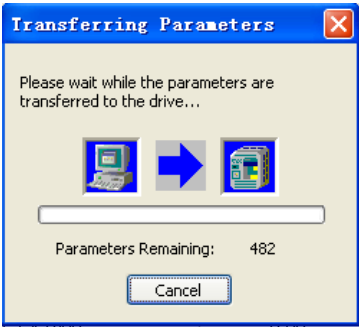


图 8.47 数据传送

习题

1. 用变频器操作面板控制变频器运行，使其连接的电动机以 5 Hz 的频率正转。
  - (1) 需要进行哪些参数初始化设置和如何进行键盘操作？
  - (2) 要想使电动机反转，需要进行哪些参数设置和键盘操作？
2. 变频器的 2 线输入和 3 线输入的最简单接线形式，如图 8.48 和图 8.49 所示。

- (1) 如何设置 A001、A002、C001、C002、C003、F002 和 F003 才能分别实现 2 线输入和 3 线输入的变频器控制。
- (2) 要求变频器输出频率如图 8.50 和图 8.51 所示，画出对应施加的开关或按钮的控制波形。
- (3) 思考在 A002 = 01 (外部电路控制变频器运行) 条件下，采用 A001 = 01 (外部电路设置频率) 或 A001 = 02 (操作面板设置频率) 两种方式设置频率是否都可以？

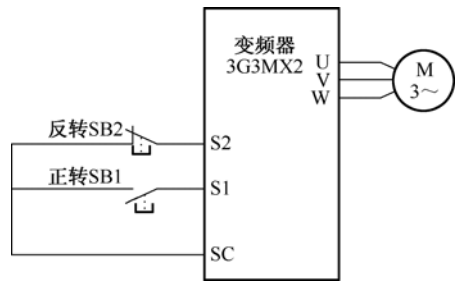


图 8.48 2 线输入接线 (省略电源线)

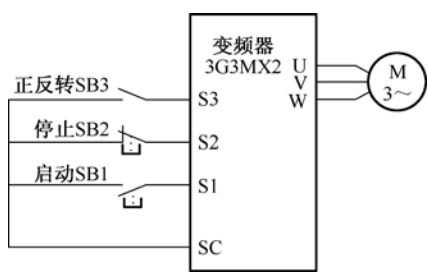


图 8.49 3 线输入接线 (省略电源线)

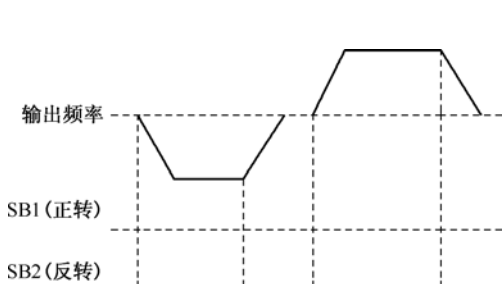


图 8.50 2 线输入控制波形

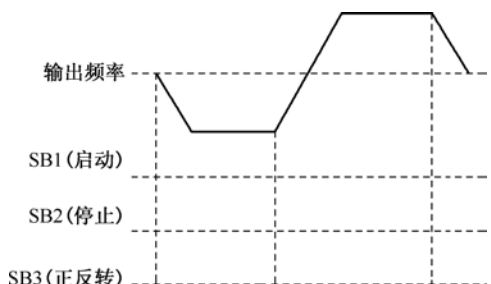


图 8.51 3 线输入控制波形

3. 某工业用洗衣机的脱水运行曲线如图 8.52 所示，一个循环分为 4 阶段。阶段 1 为 0 Hz 升到 10 Hz 脱水，整个阶段为 60 s。要求 0 Hz 升至 10 Hz 的时间为 10 s；经过慢速脱水后负载变轻，可升至阶段 2 中速脱水。阶段 2 为 25 Hz 脱水，整个阶段为 100 s。要求 10 Hz 升至 25 Hz 的时间为 5 s；阶段 3 进入快速甩干，为 40 Hz 甩干，整个阶段为 80 s。要求 25 Hz 升至 40 Hz 的时间为 5 s；阶段 4 为脱水完毕停机阶段。工作频率预置为 0 Hz，要求 40 Hz 降速为 0 Hz 时间为 20 s。试按上述要求设计 PLC 与变频器连接，设置变频器有关参数，编写 PLC 程序，并进行调试运行。

注意：F002/F202 (第 1/第 2 加速时间设定) 为设定从 0 Hz 到最高频率所需的加速时间。F003/F203 (第 1/第 2 减速时间设定) 为设定从最高频率到 0 Hz 所需的减速时间。而 A004/A204 为第 1/第 2 最高频率设定，因此要根据所设置的最高频率，选择加/减速时间设定。例如，令 A004 = 50 Hz，阶段 1 要求升至 10 Hz 的时间为 10 s，所以对阶段 1 设置加速时间为 50 s，即 0 Hz 到 50 Hz 的加速时间为 50 s，实现 10 s 由 0 Hz 升到 10 Hz。

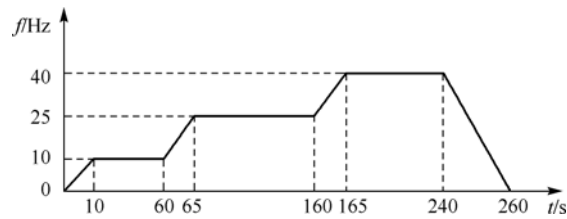


图 8.52 洗衣机输出频率图形

4. 利用 PLC 通过 Modbus RTU 通信实现变频器的多段速控制，要求变频器输出频率如图8.53所示。

- (1) 选择 PLC 并画出 PLC 与变频器的硬件连线。
- (2) 进行 PLC 设定、接口设定和变频器设定。
- (3) 编写通信程序。

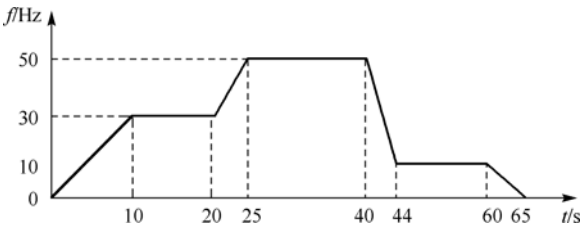


图 8.53 输出频率图形

# 第 9 章 PLC 实验

## 9.1 熟悉 PLC 及 CX-P 编程软件

### 1. 实验目的

熟悉 PLC 主机的结构，熟练运用上位机编程软件 CX-P。

### 2. 预习要求

- ① 复习第 2 章中 CP1H-XA40DR-A 主机的面板结构。
- ② 仔细阅读第 4 章 CX-P 编程软件的使用方法，预习实验内容及要求。

### 3. 实验内容和步骤

本实验装置配置一台 CP1H-XA40DR-A PLC 和一台个人计算机。要了解 CP1H PLC 主机面板各组成部分的功能及各端子、端口与外部电路的连接方法。CP1H-XA40DR-A PLC 通过 RS-232C 通信电缆或 USB 接口与 PLC 连接，在 Windows 2000、Windows XP 或更高版本的操作系统下，运行与 PLC 配套的上位机编程软件 CX-P，可直接以交互方式编写梯形图程序，在程序执行中进行监控，以及在线编辑修改等。实验所采用的方法为更为简单的直接输入法，只输入指令的地址，而没有输入地址对应的符号，不像第 6 章的符号编程方法(采用符号的编程方式)。

在桌面上双击图标，即可启动 CX-P，CX-P 窗口如图9.1所示。

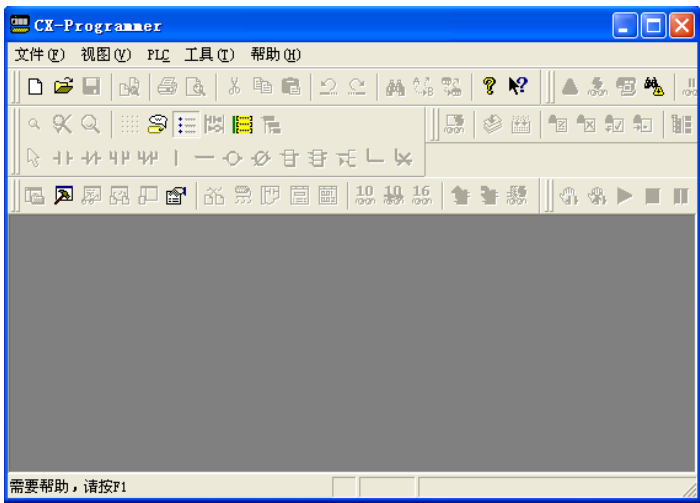


图 9.1 CX-P 窗口

#### (1) PLC 系统设定

首先进行 PLC 系统设定，设定步骤如下：

单击“文件”菜单中的“新建”命令或快捷按钮，出现如图9.2所示的“变更 PLC”对话框。



图 9.2 “变更 PLC”对话框

- ① 在“设备名称”文本框输入为 PLC 定义的名称，如输入“电机控制”。
- ② 在“设备类型”下拉列表框中选择 PLC 的系列，例如选择“CP1H”。单击“设定”按钮可进一步配置 CPU 型号，如选择“XA”。
- ③ 在“网络类型”下拉列表框中选择 PLC 的网络类型，如 USB 接口连接，选择“USB”；RS-232C 通信电缆连接，选择“SYSMAC WAY”。
- ④ 在“注释”编辑框中输入与此 PLC 相关的注释。
- ⑤ 在图9.2“变更 PLC”对话框中，单击“确定”按钮，

显示如图9.3所示 CX-P 主窗口，表明建立了一个新工程。

该窗口左面为工程工作区，右面为梯形图工作区。工程工作区中，通过显示一个与工程相关的 PLC 和程序细节的分层树状结构来表示工程。梯形图工作区可以显示梯形图程序、该程序的符号表或者助记符图。

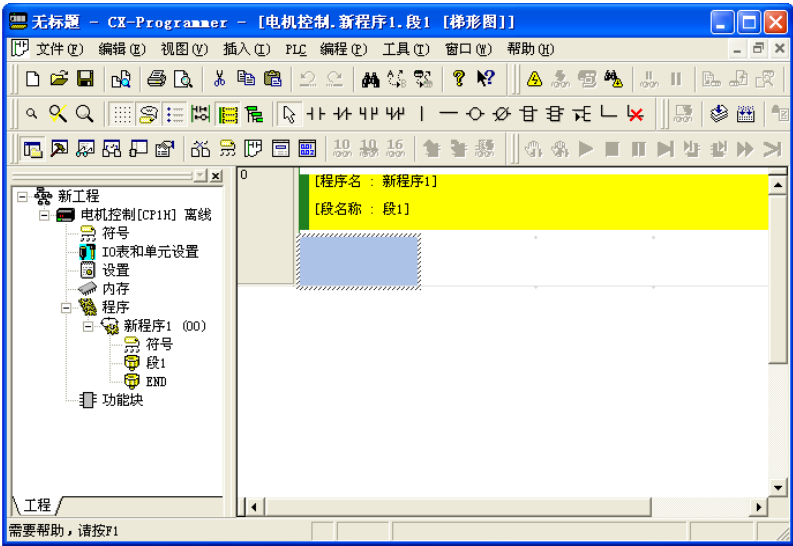


图 9.3 CX-P 主窗口

(2) 编写梯形图程序

在离线状态下创建一个梯形图程序。梯形图工具栏如图9.4所示。

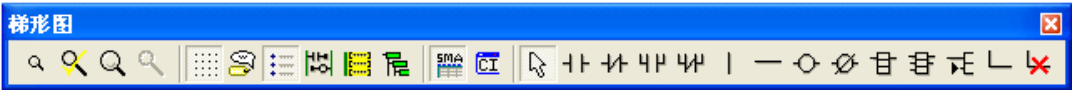



图 9.4 梯形图工具栏

在编程中常用到的是梯形图工具栏的快捷按钮为：




由左至右依次为基本指令：


选择模式、新接点、新常闭接点、新触点或、新闭合触点或、新的纵线、新的横线、新线圈、新常闭线圈、新的 PLC 指令和新功能块调用,其余特殊指令如定时器/计数器等均使用工具按钮。梯形图程序可以通过选择工具栏中的相应工具按钮,以交互方式建立,非常方便快捷。


当建立好一个新的工程时,一个空的梯形图程序将自动显示在工作区的右侧。梯形图程序的建立顺序是从左到右,从上到下。

下面,以前面讨论过的电动机启动/停止控制为例,说明程序的输入方法。电动机启动/停止控制的梯形图程序如图9.5所示。

① 工具栏中选择常开触点按钮,然后,单击第一条最左边的格子,放置一个常开触点,在弹出的对话框中输入地址“0.00”,单击确定按钮,在编辑注释对话框中输入“启动”(也可以不加注释),单击确定按钮。


此时在条的左边缘将显示一条红色线,这表明这一条出现了一个错误。当这一行输入完整后会自动消失。

② 在工具栏中选择常闭触点按钮,在常开触点右边放置一个常闭触点,输入地址“0.01”,单击确定按钮,在编辑注释对话框中输入“停止”(也可以不加注释),单击确定按钮。

③ 在工具栏中选择线圈按钮,在常闭触点的右边放置一个线圈,输入地址“100.00”,单击确定按钮,编辑注释“输出驱动”,单击确定按钮。

此时,在条的左边缘已不再显示红色线,因为在这个条里已经没有错误了。

当选择的格子在条的右边时,按回车键,建立一个新行。

④ 在工具栏中选择并联的常开触点按钮,在常开触点 0.00 的下方放置一个新的常开触点(它和常开触点 0.00 还在同一个条里),输入地址 100.00。

建立好的梯形图程序如图9.6所示。

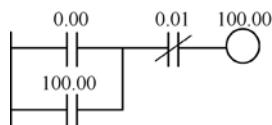


图 9.5 电动机启动/停止控制梯形图程序

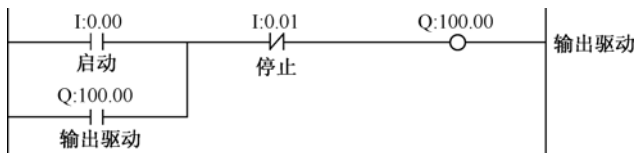


图 9.6 启动/停止控制梯形图在 CX-P 中的视图


### (3) 编译

当程序编写完成时,在“编程”菜单中选择“编译”命令,系统开始对整个程序进行检查,并自动列出程序中所有的错误细目,显示在输出窗口的编译标签下面。如果仅给出警告,可以不予理会,这是允许的,如果给出错误,这是不允许的,必须进行查错并修改之。可以按照错误细目对程序进行编辑与修改。

无论是在线程序还是离线程序,在其生成和编辑过程中都会不断被检验,并在梯形图中,将程序错误以红线表示(如果条中出现一个错误,在梯形图条的左边将会出现一道红线)。

### (4) 把程序传送(下载)到 PLC 并开始运行

从此步开始需要 PLC 在线工作。

单击工具栏中的保存工程按钮,保存当前的工程。如果此前还未保存过,即会显示保存 CX-P 文件对话框,输入文件名称,然后单击保存按钮,完成保存操作。

① 接通 PLC 主机上的电源。




② 在“PLC”菜单中选择“在线工作”命令,或单击工具栏中的“在线工作”按钮,在弹出



图 9.7 对话框

的对话框(如图9.7 所示)中单击“是”按钮，由于在线时一般不允许编辑，所以程序变成灰色。

③ 在“PLC”菜单中选择“操作模式”→“编程”命令，或直接单击工具栏中的“编程模式”按钮，在弹出的对话框中单击“是”按钮，把 PLC 的操作模式设为编程。

④ 在“PLC”菜单中选择“传送”→“到 PLC”命令，在弹出的“下载选项”对话框中选择“程序”，并单击“确定”按钮，或直接单击工具栏中的“传送到 PLC”按钮，将显示“下载选项”对话框，可以选择的项目有程序、设置、特殊单元设置、符号、注释等。按照需要选择后，单击“确定”按钮。

接下来的出现的一系列动作，取决于 PLC 原来的工作状态，例如原工作状态为“监视模式”。

⑤ 在接下来的几个对话框“此命令将影响所连接的 PLC 状态要继续吗？”(请选择：是)、“想要切换到编程模式？”(请选择：是)、下载(请选择：确定)、“想要切换到监视模式？”(请选择：是)。

程序传送到 PLC 并开始运行，对输入口进行巡回检测，等待用户的操作命令，若 100.00 为三相异步电动机交流接触器的线圈通断电控制。按下启动按钮 0.00，100.00 通电，电动机即开始运行，按下停止按钮 0.01，电动机即停止运行。此操作可以反复进行。

#### (5) 在程序执行中进行监视

一旦程序被下载，就可以在程序工作区中对其运行监视(以模拟显示的方式)。即程序执行时，可以监视梯形图中的数据和控制流，例如，连接的选择和数值的增加。

按照以下步骤来监视程序：

在“PLC”菜单中选择“操作模式”→“监视”命令，或单击工具栏中的“切换 PLC 监视”按钮




数据监视值的格式将取决于被监视的用做操作数的符号或者指令操作数的数据类型。



#### (6) 在线编辑


在程序运行过程中可能需要修改，但已经下载的程序会变成灰色，以防止被直接编辑，不过，可以选择在线编辑功能来修改梯形图程序，此时，必须使 PLC 运行在监视模式下，在运行模式下是不可能进行在线编辑的。


使用以下步骤进行在线编辑：

① 拖动鼠标，选择要编辑的条。

② 在“PLC”菜单中选择“传送”→“与 PLC 比较”命令，或直接在工具栏中单击“与 PLC 进行比较”按钮，以确认编辑区域的内容和 PLC 内的相同。

③ 在“编程”菜单中选择“在线编辑”→“开始”命令，或直接在工具栏中单击“在线编辑条”按钮，要编辑的条的背景将改变为白色，表明其现在已经是一个可编辑区。此区域以外的条的颜色不能被改变，但是可以把这些条里面的元素复制到可编辑条中去。如果光标在可编辑条区域以外，可单击“转到在线编辑条”按钮，则返回到在线编辑条的最上面。

④ 把光标移到需要修改的条上进行在线修改。结束编辑修改时，在“编程”菜单中选择“在线编辑”→“发送修改”命令，或直接在工具栏中单击“发送在线编辑修改”按钮，所编辑的内容将被检查并且被传送到 PLC。一旦这些改变被传送到 PLC，编辑区域再次变成只读。

如果想取消在线编辑状态，可以在“编程”菜单中选择“在线编辑”→“取消”命令，或者单击工具栏中的“取消在线编辑”按钮。

注意，在线编辑时不能改变符号的地址和类型。



(7) 练习

输入并调试第 3 章中图 3.20~图 3.30，熟悉 CX-P 编程软件。

4. 实验报告内容

- ① 在 CX-P 编程软件中，对 PLC 进行初始设置应包括哪些内容？
- ② 本例实验，采用直接输入指令和地址编程方式。如果采用符号编程，需要如何做？

9.2 定时器/计数器指令

1. 实验目的

进一步熟悉 CX-P 编程软件的使用。掌握输入/输出、定时器/计时器、微分、保持等常用指令的功能和编程方法。

2. 预习要求

- ① 复习教材中与本次实验有关的指令。
- ② 分析实验内容中各程序段的执行结果，弄清各实验内容的目的。

3. 实验内容和步骤

(1) 基本逻辑指令练习

- ① 输入如图9.8所示的程序，运行程序，观察程序执行结果并填在表中。
- ② 输入如图9.9所示的程序，运行程序，观察各输入点的作用及自保持现象。

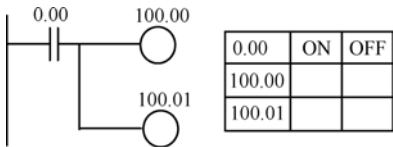


图 9.8 ①图

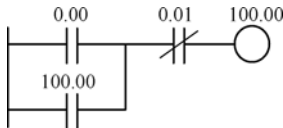


图 9.9 ②图

- ③ 输入如图9.10、图9.11所示的程序，运行程序，观察各输入点的作用及输入、输出 LED 的状态。

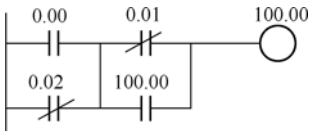


图 9.10 串联触点组并联程序

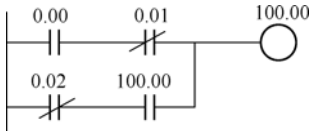


图 9.11 并联触点组串联程序

(2) TIM/CNT 指令练习

- ① 输入如图9.12所示的程序。
- ② 自闭合自锁开关 0.00 起，监视 CNT0001 的计数和 TIM0000 定时的全过程。
- ③ 将 CNT0001 的设定值改为#0005，监视 CNT0001 的计数过程。在其当前值为 0002 时，断开电源，观察复电后 100.00 对应的 LED 过几秒时亮。分析计数器是否有掉电保持功能。
- ④ 用 TIM 指令编写一个定时程序，用 100.00 观察其输出。当 100.00 对应的 LED 亮后断电再复电，分析定时器是否有掉电保持功能。

(3) CNTR 指令练习

- ① 输入如图9.13所示的程序。

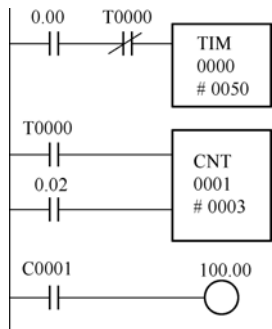


图 9.12 TIM/CNT 指令程序

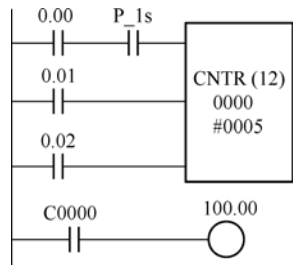


图 9.13 CNTR 指令程序

- ② 闭合自锁开关 0.00，断开 0.01 后运行程序。监视 CNTR0000 的计数过程，并注意 100.00 的状态。
- ③ 将 CNTR 复位，按上述条件重新运行程序，监视 CNTR0000。在其当前值为 0002 时断开电源，观察复电后 100.00 过几秒时亮。
- ④ 修改程序，将图 9.13 中 P\_1s 与 0.01 串联，闭合自锁开关 0.01，断开 0.00 后运行程序。监视 CNTR0000 的计数过程，注意观察与步骤②计数过程的区别。

(4) DIFU、DIFD 指令练习

- ① 输入如图 9.14 所示的程序。
- ② 运行程序，接通开关 0.00 时，观察 100.00 和 100.01 的状态；断开开关 0.00 时观察 100.00 和 100.01 的状态。
- ③ 将 100.00 和 100.01 复位。重新做一次步骤②的内容。
- ④ 将 PLC 断电，先将开关 0.00 闭合后再接通电源，观察 100.00 和 100.01 的状态。
- ⑤ 将 PLC 断电，先断开开关 0.00 后再接通电源，观察 100.00 和 100.01 的状态。

(5) KEEP 指令练习

- ① 输入如图 9.15 所示的程序。
- ② 运行程序，在断开 0.00 和闭合 0.00 时，观察 100.00 的状态。
- ③ 将图 9.15 中 P\_1s 改为 0.01。运行程序，当 100.00 为 ON 时断开 0.00，观察 100.00 的状态；接着再断开 PLC 电源，观察再复电时 100.00 的状态。
- ④ 将 P\_1s 改为 0.01，100.00 改为 H1.00。运行程序，当 H1.00 为 ON 时先断开 0.00，再断开电源，监视复电时 H0.00 的状态。

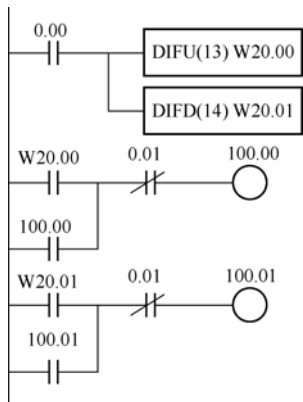


图 9.14 DIFU、DIFD 指令程序

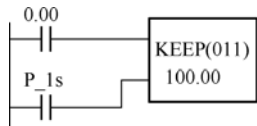


图 9.15 KEEP 指令程序

#### 4. 实验报告内容

- ① 总结 CNT 和 CNTR 的区别。
- ② DIFU 和 DIFD 指令有何用途, 使用它们时应注意什么问题?
- ③ 总结 KEEP 指令的功能。

### 9.3 三相异步电动机的 PLC 控制

#### 1. 实验目的

练习使用 PLC 实现各种控制, 加深理解用 PLC 进行控制的原理, 并了解 PLC 控制与继电器控制的区别。

#### 2. 预习要求

- ① 根据每个实验内容的要求, 选择输入/输出电器, 做出 I/O 分配表, 画出 PLC 的外部连线图。
- ② 根据每个实验内容的要求, 画好梯形图, 写出语句表。
  - 用 PLC 实现对三相异步电动机的直接启、停控制, 要求有短路、零压和过载保护。用两个按钮进行启、停操作。为了模拟电动机过载, 再用一个按钮的触点代替热继电器的触点。
  - 用 TIM 指令编写程序, 实现白炽灯先亮, 延时 5 s 后电动机自行启动的控制。要求用两个按钮进行启、停操作。
  - 编写电动机正反转的控制程序。要求正反转要互锁, 电动机正转过程中欲反转, 必须先停机才能反转。用 3 个按钮进行启停、正反转操作。
- ③ 画出各实验内容中电动机等负载的主电路图。

#### 3. 实验内容和步骤

##### (1) 三相异步电动机的直接启、停控制

- ① PLC 断电, 按 I/O 分配表将各电器与主机连接好, 接好电动机的主电路。
- ② 给 PLC 再加电, 将写好的程序输入到 CX-P 软件, 并下载到 PLC。
- ③ 将工作模式换到监控模式, 进行下列操作, 并记录现象。
  - 按启动按钮, 观察电动机的启动。电动机运行途中按停止按钮, 观察停止按钮的作用。
  - 按下模拟热继电器的按钮(模拟电动机过载), 观察电动机是否停机。
  - 重新启动电动机。在电动机运行途中切断电源(模拟断电), 待电动机停止转动后, 再接通电源, 观察电动机是否能自行启动。

##### (2) 白炽灯与电动机的联锁控制

- ① PLC 断电, 按 I/O 分配表将各电器与 PLC 主机连接好, 接好灯和电动机的主电路。
- ② 给 PLC 再加电, 将编写的程序下载到 PLC。
- ③ 按启动按钮, 观察电动机和灯的状态是否符合设计要求。
- ④ 将 TIM 的定时时间改为 8 s, 运行程序, 观察电动机和灯的状态。
- ⑤ 修改程序, 令灯先亮 5 s 后电动机再启动, 运行程序, 观察电动机和灯的状态。

##### (3) 电动机正、反转控制

- ① PLC 断电, 按 I/O 分配表将各电器与 PLC 主机连接好, 接好电动机的主电路。
- ② PLC 加电, 将写好的程序下载到 PLC。
- ③ 将工作模式换到监控模式, 进行下列操作, 并记录现象。

- 按正转按钮，观察电动机的转向。正转过程中按反转按钮，观察电动机的转向是否改变。
- 按下停车按钮，待电动机停止转动后，再按反转按钮，观察电动机的转向是否改变。反转过程中按下正转按钮，观察电动机的转向是否改变。

4. 实验报告内容

- ① 画出各实验内容所需要的梯形图。
- ② 写出各实验内容的 I/O 分配表。
- ③ 画出各实验内容 PLC 的外部接线及负载的主电路图。
- ④ 用继电器接触器和 PLC 都可以对电动机进行实验所要求的控制，总结两种控制的区别。

9.4 互锁/跳转/数据比较/数据移位/数据传送指令

1. 实验目的

理解互锁、跳转、数据比较、数据移位、数据传送指令的功能，掌握其使用方法。

2. 预习要求

- ① 复习与本次实验相关的指令。
- ② 阅读实验程序，分析运行结果，弄清各实验的目的。
- ③ 编写实验内容中要求的部分程序。
- ④ 会用编程软件向通道写入数据。
- ⑤ 会用编程软件进行通道监视、CNT/TIM 监视和位监视。

3. 实验内容和步骤

(1) IL/ILC、JMP/JME 指令练习

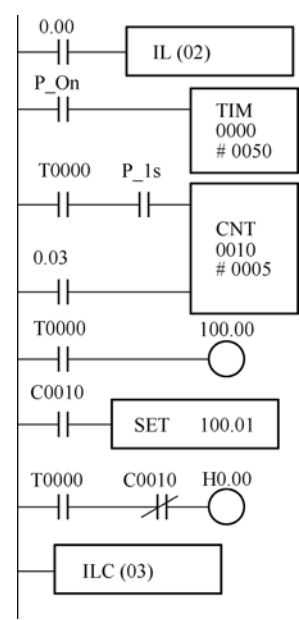


图 9.16 IL/ILC、JMP/JME 指令程序

- ① 输入如图9.16所示的程序。
- ② 运行程序，接通 0.00 后观察程序执行的全过程。注意 100.00、100.01 的状态。
- ③ 继续步骤②，断开 0.00，观察 100.00 和 100.01 的状态。监视并记录 H0.00、CNT0010 和 TIM0000 的内容。

- ④ 将图9.16程序中的 IL/ILC 换成 JMP/ JME，重做步骤②~③的内容。

(2) 数据传送指令练习

① MOV、MVN 指令的练习。

- 输入如图9.17所示的程序。
- 运行程序，监视 W20 通道的内容并填入表中。
- 将图9.17中的 MOV 指令改为 MVN，运行程序，监视 W20 通道的内容，并与 MOV 指令时表的内容比较。

② BSET 指令练习。

- 输入如图9.18所示的程序。
- 运行程序，令 0.00 为 ON，0.01 为 OFF，监视 W20 和 TIM0000 的内容。

- 再次运行程序, 令 0.00 为 ON, 0.01 为 OFF, 监视 W20 和 TIM0000 的内容。在 TIM0000 定时过程中, 闭合一次 0.01 注意观察并记录 W20 和 TIM0000 的内容是否改变, 怎样变。注意 BSET 指令是改变了 TIM0000 的当前值还是设定值。

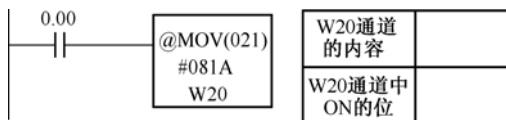


图 9.17 MOV、MVN 指令程序

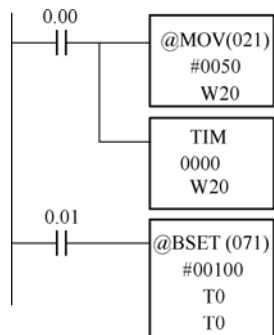


图 9.18 BSET 指令程序

- 用 MOV 指令能否向 TIM0000 中传送数据? 编写一个程序, 试输入程序。
- 用 BSET 指令能否改变 TIM0000 的设定值? 编写一个程序, 通过实验验证其是否可行。

### (3) 数据移位指令练习

#### ① SFT 指令练习。

- 输入如图9.19所示的程序。
- 闭合 0.00, 断开 0.02, 用按钮 0.01 不断输入移位脉冲, 观察并记录 100 通道各位的状态。
- 将移位寄存器复位。断开 0.02, 闭合 0.00, 用按钮 0.01 输入两个移位脉冲时, 断开 0.00 且继续输入移位脉冲。
- 观察并记录 100 通道各位的状态。

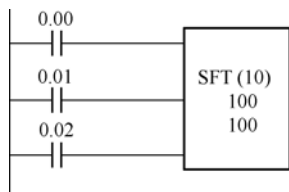


图 9.19 SET 指令程序

将移位寄存器复位。断开 0.02, 闭合 0.00, 当输入两个移位脉冲时, 断开电源再复电, 观察并记录 100 通道各位的状态与断电前是否有变化。

- 将图9.19中的 100 改为 H0, 按上一步的步骤, 监视 H0 各位的状态与断电前是否有变化。

#### ② SFTR 指令练习。

- 输入如图9.20所示的程序。
- 运行程序前, 令 0.00、0.01 为 ON, 令 0.03 为 OFF, 令 0.02、0.04 为 ON, 观察并记录 100 通道各位的状态; 途中断开 0.01, 再观察并记录 100 通道各位的状态。
- 令 0.03 为 ON, 将移位寄存器复位, 观察并记录 100 通道各位的状态。
- 复位寄存器, 再次运行程序。先令 0.00 为 OFF, 0.01 为 ON, 0.03 为 OFF, 再令 0.02、0.04 为 ON, 观察并记录 100 通道各位的状态; 途中断开 0.01, 再观察并记录 100 通道各位的状态。
- 复位寄存器, 再次运行程序。先令 0.00 为 OFF, 0.01、0.03 为 ON, 再令 0.02、0.04 为 ON, 观察并记录 100 通道各位的状态。
- 将图中 P\_1s 所在行去掉, 将 W20.14 的控制触点改为 0.02。先令 0.00、0.01 为 ON, 0.03 为 OFF。再令 0.04 为 ON。运行程序, 观察并记录 100 通道各位的状态。
- 在第 4 步的条件下运行, 途中断开 PLC 的电源。再复电时观察并记录 100 通道各位的状态。

#### ③ SLD 指令练习。

- 输入如图9.21所示的程序。

- 运行程序，闭合一次 0.00，监视 W20 和 W21 通道的内容。
  - 闭合一次 0.01，监视 W20 和 W21 通道的内容。
  - 再闭合几次 0.01，监视 W20 和 W21 通道的内容。
- (4) 数据比较指令练习
- ① CMP 指令练习。
- 设计一个程序，比较 W20 和 W21 通道中数据(W20 和 W21 通道中的数据自定大小)，并通过 100 通道来观察比较结果，画出梯形图，输入程序。

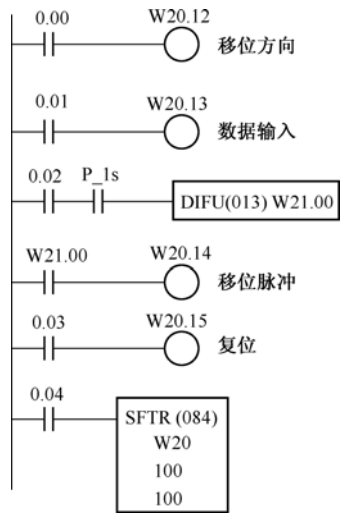


图 9.20 SFTR 指令程序

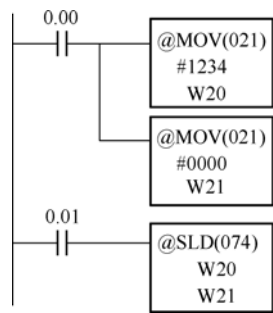
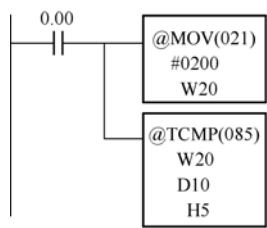


图 9.21 SLD 指令程序

- 上机调试程序并观察并记录下列各种情况的运行结果(观察 100 通道各位的状态)：  
W20 通道的内容 = W21 通道的内容；  
W20 通道的内容 > W21 通道的内容；  
W20 通道的内容 < W21 通道的内容。
- ② TCMP 指令练习。
- 输入如图9.22 (a) 所示的程序。



(a)

D10	0100	H5.00	
D11	0200	H5.01	
D12	0300	H5.02	
D13	0400	H5.03	
D14	0500	H5.04	
D15	0600	H5.05	
D16	0200	H5.06	
D17	0800	H5.07	
D18	0100	H5.08	
D19	1000	H5.09	
D20	0200	H5.10	
D21	1200	H5.11	
D22	1300	H5.12	
D23	1400	H5.13	
D24	1500	H5.14	
D25	1600	H5.15	

(b)

图 9.22 TCMP 指令程序

- 按图9.22(b)中的内容向 D10~D25 写入数据。
- 运行程序, 监视 H5 通道的内容(用位监视), 将监视的结果填在图9.22(b)中。

#### 4. 实验报告内容

- ① 总结 IL/ILC 与 JMP/JME 的区别。
- ② 在 BSET 指令练习时, 第3步中 BSET 指令是改变 TIM0000 的当前值还是设定值; 能否用 MOV 指令向 TIM0000 中传送数据? 用 BSET 指令能否改变 TIM 的设定值? 请写出相关的程序。
- ③ 请写出 SFT 指令练习中第2~4步的现象, 用 H0 做移位寄存器是否有掉电保持?
- ④ 整理 SFTR 指令练习中步骤第2~7步的现象, 并指出 SFTR 与 SFT 指令的区别。
- ⑤ 画出比较 W20 和 W21 通道数据大小的梯形图, 整理比较结果(用 100 通道的某些位的状态表示)。

## 9.5 彩灯的 PLC 控制

### 1. 实验目的

进一步熟悉移位寄存器等指令的功能和使用方法, 进一步了解互锁和跳转指令的区别。

### 2. 预习要求

- ① 仔细阅读教材中移位寄存器、互锁、跳转等指令的功能、执行条件和编程方法。
- ② 仔细阅读参考程序(如图9.23所示), 弄清楚程序的功能。参考程序说明如下:
  - 0.01、0.02、0.03、0.04 为自锁开关, 0.01~0.03 用来选择灯亮的个数及编排花样, 0.04 选择亮灯的移动方向。
  - 当 0.04 输入点的开关断开时, 亮灯从左向右移动(100.00~100.07), 当移到最右侧后, 重复上述移动过程。
  - 当 0.04 输入点的开关闭合时, 亮灯从右向左移动(101.07~101.00), 当移到最左侧后, 重复上述移动过程。
  - TIM0000 可以改变移位寄存器移位的速度。

### 3. 实验内容和步骤

#### (1) 输入程序

输入如图9.23所示的参考程序。经检查确定输入的程序无误, 下载到 PLC 后, 将工作模式切换到监视模式。

#### (2) 运行程序

- ① 观察彩灯移动方向。令 0.04 输入点的开关断开, 且 0.01 闭合, 0.02、0.03 断开。接通 0.00 后程序开始执行, 观察并记录灯亮的个数与移动方向。途中接通 0.04 输入点的开关, 观察并记录 100、101 通道的状态。
- ② 观察彩灯花样的变化(0.04 输入点的开关断开)。
  - 断开 0.00, 令 0.01 和 0.02 闭合, 0.03 断开。接通 0.00 程序开始执行, 观察并记录灯亮的个数与移动方向。
  - 断开 0.00, 令 0.01 和 0.03 闭合, 0.02 断开。接通 0.00 程序开始执行, 观察并记录灯亮的个数与移动方向。
- ③ 观察彩灯移动速度的变化(0.04 输入点的开关接通)。

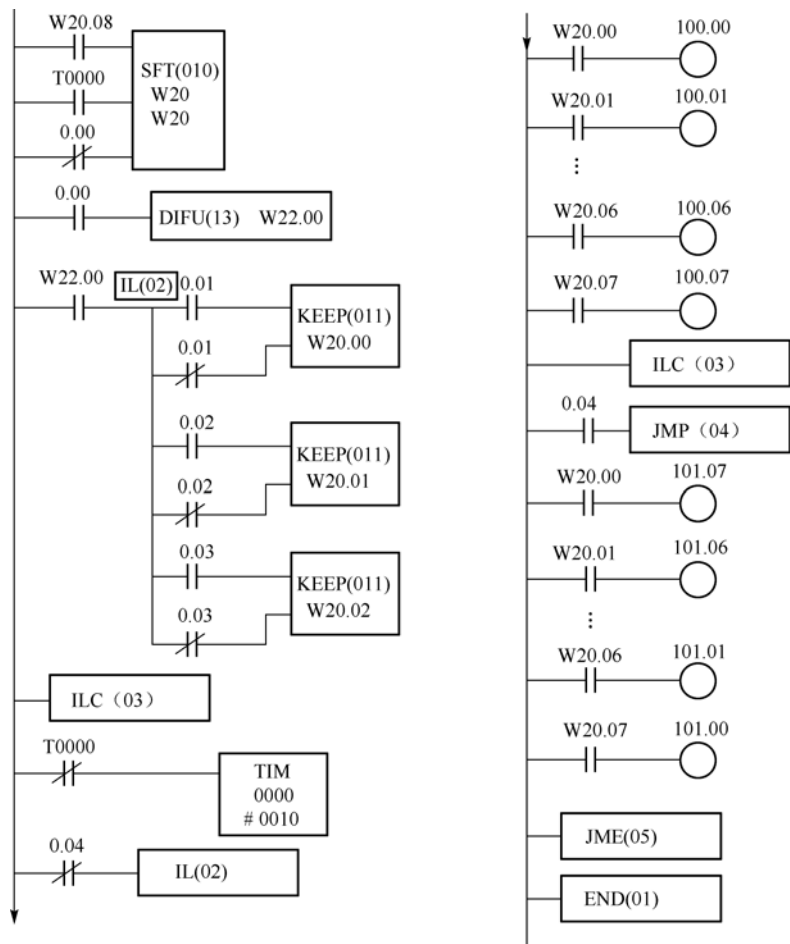


图 9.23 实验程序

- 将 TIM0000 的定时值改为#0020。按步骤②中的第 2 步操作，运行程序，观察并记录亮灯的移动速度有无变化。
- 途中断开 0.04 输入点的开关，观察并记录 101 通道的状态。

4. 实验报告内容

- ① 简述图9.23参考程序的功能，指出 TIM0000 的作用。
- ② 分析互锁指令、跳转指令之间的区别。

9.6 数据转换及数据运算指令

1. 实验目的

理解数据转换、十进制运算、二进制运算、逻辑运算等指令的功能，掌握其使用方法。

2. 预习要求

- ① 仔细阅读教材中有关数据转换、十进制运算、二进制运算、逻辑运算等指令的内容，熟悉各指令的功能、数据范围和执行条件。



② 仔细阅读各参考程序，分析各程序的运行结果，弄清各实验的目的。

3. 实验内容和步骤

(1) 数据转换指令练习

① BCD 码→二进制转换指令 BIN 练习。

- 输入如图9.24所示的程序。
- 运行程序，闭合 0.00，监视并记录 W20 和 H1 通道的内容。

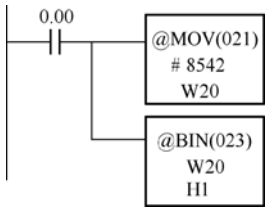


图 9.24 BIN 指令程序

② 二进制→BCD 码转换指令 BCD 练习。

- 输入如图9.25所示的程序，@MOV 指令的数据 SV 是步骤①H1 通道中的转换结果。

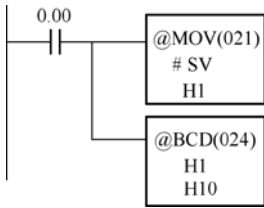


图 9.25 BCD 指令程序

- 运行程序，监视并记录 H1 和 H10 通道中的内容。

③ 译码 MLPX 指令练习。

输入如图9.26(a)所示的程序。

执行@MLPX 指令后，监视并记录 H0、H1、H2、H3 通道中的数据。

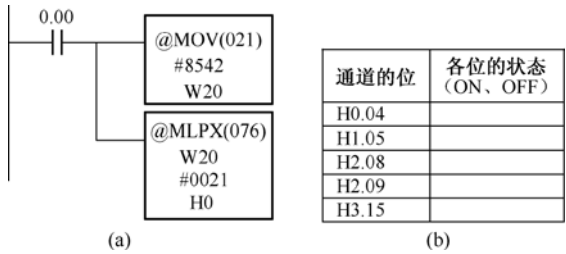


图 9.26 MLPX 指令程序

④ 七段译码 SDEC 指令练习。

- 输入如图9.27所示的程序。
- 执行@SDEC 指令后，观察并记录 100 通道的输出状态，即观察输出 LED 亮的位置与七段数码管对应段是否相符。七段数码管见第 3 章图 3.63。

如果将图 9.27 的控制字改为#0011 时，需要几个七段数码管？每个七段数码管显示的数码是什么？

⑤ DMPX 指令练习。

- 输入如图 9.28 所示的程序。
- 运行程序，监视 W20、W21 通道中的数据，指出两通道中的最高位。
- 执行@DMPX 指令后，监视并记录 H9 通道的内容。

(2) 十进制运算指令练习

① +BC(406)指令练习。

- 输入如图9.29所示的程序，向 H15 通道写入常数#8766。
- 运行程序，监视并记录 H15、H9 通道中的数据，观察并记录 100.00 的状态。

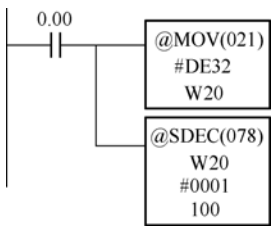


图 9.27 SDEC 指令程序

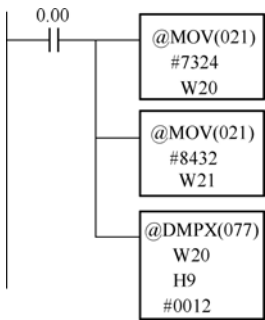


图 9.28 DMPX 指令程序

- 欲保存 P\_CY 的内容，请修改如图9.29所示的程序，运行程序，观察并记录结果。
  - 欲进行双字加运算时程序应怎样编写？要占用几个通道？请设计一个程序，运行程序，并观察记录各通道的内容。
- ② -BC(416)指令练习。
- 输入如图9.30所示的程序。
  - 运行程序，监视并记录 H0 的内容。
  - 将程序中的被减数改成 0766，再运行程序，监视并记录的 H0 内容。
  - 欲进行双字减法运算时程序应怎样编写？要占用几个通道？请设计一个程序，并运行程序、观察记录各通道的内容。

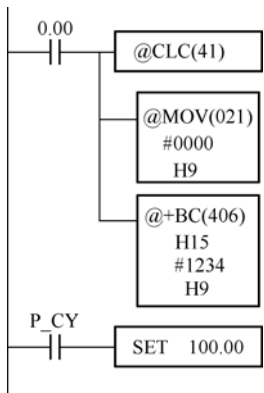


图 9.29 +BC 指令程序

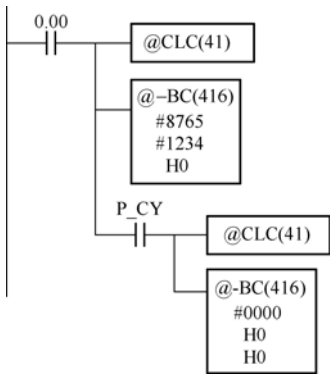


图 9.30 -BC 指令程序

- ③ \*B(424)、/B(434)指令练习。
- 输入如图9.31所示的程序。
  - 先将 D0~D6 清零，向 D0 通道写入#1234，向 D1 通道写入#5231，向 D4 通道写入#5245，向 D5 通道写入#2254。
  - 运行程序，指出乘法运算结果的高位和低位各存在哪个通道。用通道监视、观察并记录乘法运算结果。
  - 指出除法运算结果的商和余数各存在哪个通道，用通道监视并记录除法运算结果。
  - 欲进行双字乘法运算，程序应怎样编写？要占用几个通道？请设计一个程序，并运行程序、观察记录各通道的内容。
- (3) 二进制运算指令练习
- ① +C(402)指令练习。

- 输入如图9.32所示的程序。
- 向 D10 通道写入数据#301A。
- 运行程序，监视并记录 D10、D11 通道中的数据，观察 100.00 的状态。

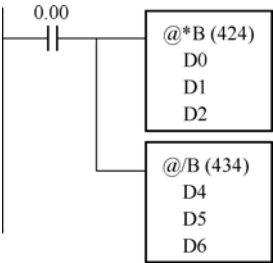


图 9.31 \*B、/B 指令程序

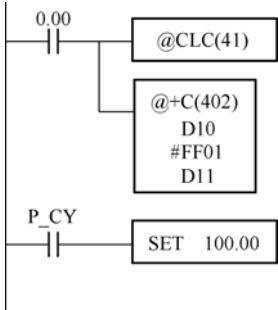


图 9.32 +C 指令程序

② \*U(422)、/U(432)、DVB(53)指令练习。

- 输入如图9.33所示的程序。
- 向 D0 写入#120E，向 D1 写入#018B，向 D10 写入#54A1，向 D11 写入#001F。
- 运行程序，指出乘法运算结果的高位和低位各存在哪个通道，监视通道并记录乘法运算结果。
- 运行程序，指出除法运算结果的商和余数各存在哪个通道，监视通道并记录除法运算结果。

(4) 逻辑运算指令

① ANDW (034)、ORW (035)指令练习。

- 输入如图9.34所示的程序。
- 分别向 D100 和 D101 写入#8463 和#5631，向 D110 和 D111 写入#8463 和#5631。
- 运行程序，监视并记录 D102 和 D112 通道各位的内容。
- 用 ANDW 指令编写一个程序，保留 H0 通道中的低 4 位的状态，屏蔽其余的位。运行程序，监视并记录 H0 中各位的内容。

② XORW (036)和 XNRW (037)指令练习。

- 输入如图9.35所示的程序。
- 分别向 D200 和 D201 写入#8463 和#5631，向 D210 和 D211 写入#8463 和#5631。
- 运行程序，监视并记录 D202 和 D212 通道各位的内容。

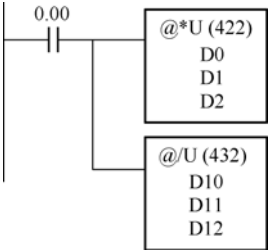


图 9.33 \*U、/U 指令程序

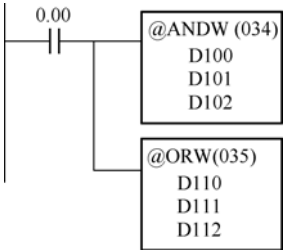


图 9.34 ANDW、ORW 指令程序

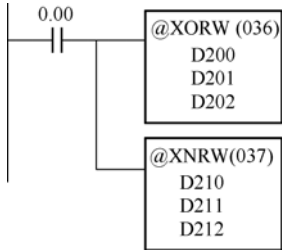


图 9.35 XORW、XNRW 指令程序

4. 实验报告内容

- ① 整理数据转换指令练习的表格和数据。
- ② 对十进制运算指令使用练习做下列报告。
- 整理所有实验结果。

- 双字加运算时程序应怎样编写？要占用几个通道？画出经上机通过的梯形图程序，并记录各通道的内容。
  - 欲进行双字减法运算时程序应怎样编写？要占用几个通道？请设计一个程序，画出经上机通过的梯形图程序，并记录各通道的内容。
  - 欲进行双字乘法运算时程序应怎样编写？要占用几个通道？画出经上机通过的梯形图程序，并记录各通道的内容。
- ③ 整理二进制运算指令练习的所有实验结果。
- ④ 整理出逻辑运算指令练习的所有实验结果。

9.7 十字路口交通灯控制

1. 实验目的

- ① 熟悉 CX-P 编程软件。
- ② 进一步熟悉基本指令的功能及应用。
- ③ 熟悉十字路口交通灯控制的编程及调试。

2. 控制要求

图9.36是城市十字路口交通灯示意图，交通灯的控制要求如下：

在十字路口上设置的红、黄、绿交通信号灯，其布置如图9.36所示。由于东西方向的车流量较小，南北方向的车流量较大，所以南北方向的放行(绿灯亮)时间为 30 s，东西方向的放行时间(绿灯亮)为 20 s。当在东西(或南北)方向的绿灯灭时，该方向的黄灯与南北(或东西)方向的红灯一起以 5 Hz 的频率闪烁 5 s，以提醒司机和行人注意。闪烁 5 s 之后，立即开始另一个方向的放行。用两个控制开关启动控制开关和停止控制开关对系统进行起停控制，要求采用顺序控制方式编写控制程序。

3. 程序设计

- ① 进行 PLC 的 I/O 分配。使用 CPIH 时所做的 I/O 分配见表 9.1。

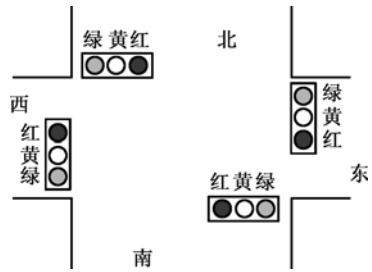


图 9.36 交通灯布置图

表 9.1 I/O 分配

输 入		输 出	
启动开关	0.00	南北绿灯	100.00
停止开关	0.01	南北黄灯	100.01
—	—	南北红灯	100.02
—	—	东西绿灯	100.03
—	—	东西黄灯	100.04
—	—	东西红灯	100.05

- ② 画顺序功能图。顺序功能图如图9.37所示，简单含义如图9.38所示。
- ③ 参考程序如图9.39 所示，采用的是顺序功能图设计法，与第 6 章的采用顺序控制设计的交通灯控制程序图 6.31 是一样的。

4. 实验报告内容

- ① 试回答 W10.00 的作用，考虑在顺序控制中如何加的启动按钮和停止按钮。
- ② 与第 6 章采用时序图设计法的交通灯控制程序图 6.8 相比，各有什么特点？

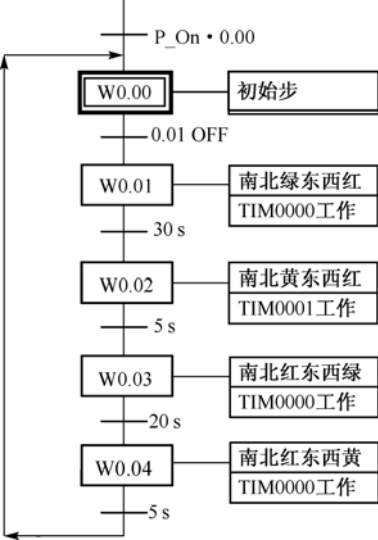


图 9.37 顺序功能图

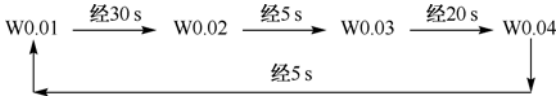


图 9.38 步的含义

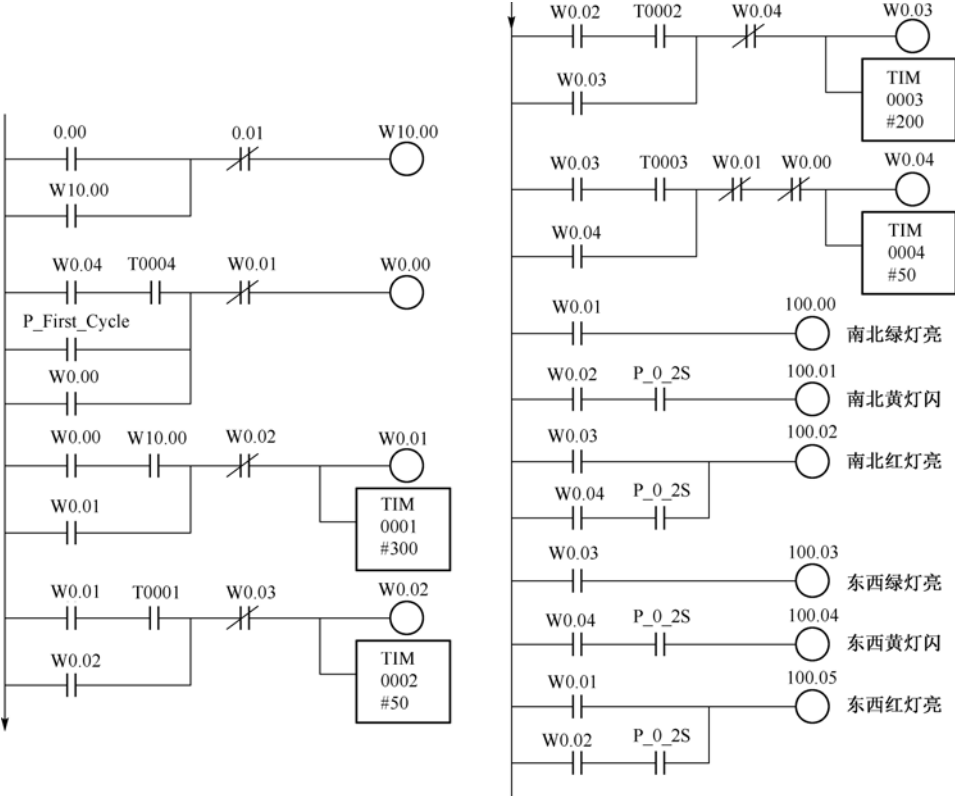


图 9.39 梯形图程序

③ 上述实验为交通灯正常时序控制，若改为急车强通控制，试编制程序。急车强通控制的要求如下：

急车强通信号受两路急车强通开关控制。无急车时，交通灯按正常时序控制。有急车来时，将急车强通开关接通，不管原来交通灯的状态如何，一律强制让急车方向的绿灯亮，使急车放行，直到急

车通过为止。急车一过，将急车强通开关断开，交通灯的状态立即转为急车放行方向上的绿灯闪 3 次，随后按正常时序控制。

## 9.8 子程序指令与中断控制指令

### 1. 实验目的

加深对子程序控制指令、中断控制指令功能的理解，能熟练用以上指令编写程序。

### 2. 预习要求

- ① 仔细阅读有关子程序控制指令、中断控制指令的内容，熟悉各指令的功能、数据范围和执行条件。
- ② 仔细阅读各参考程序，并弄清各程序的功能和程序运行结果。

### 3. 实验内容和步骤

#### (1) 子程序控制指令使用练习

- ① 输入如图9.40所示的参考程序。

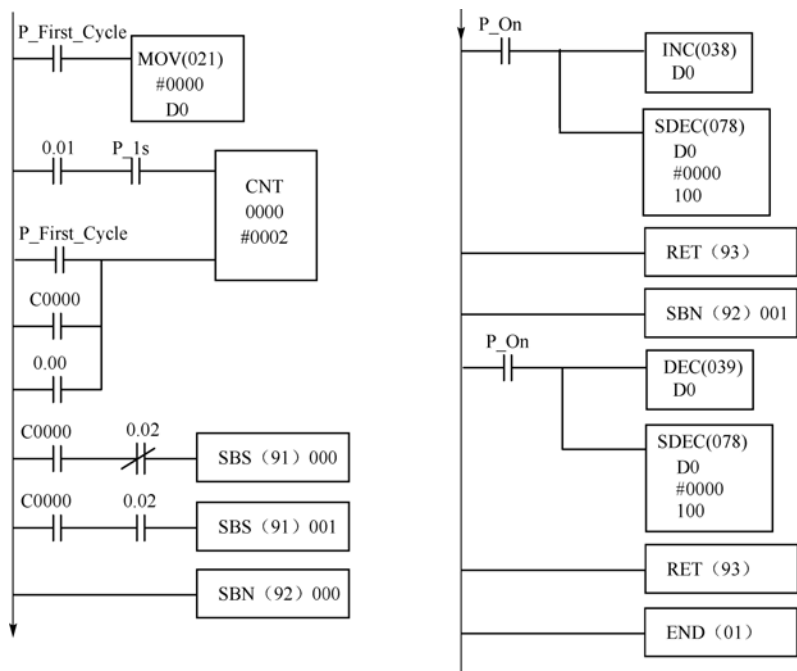


图 9.40 子程序参考程序

- ② 按下面的要求运行程序，监视并记录相关数据。

- 0.02 OFF 时运行程序，监视并记录通道 D0 中数据的变化(记录 5 次变化的数据)。
- 0.02 OFF 时运行程序，观察并记录执行 SDEC 指令后 100 通道输出的状态。
- 0.02 ON 时运行程序，监视并记录通道 D0 中数据的变化(记录 5 次变化的数据)。
- 0.02 ON 时运行程序，观察并记录执行 SDEC 指令后 100 通道输出的状态。

#### (2) 输入中断控制指令使用练习

- ① 在 CX-P 编程软件的工程目录，双击项目树的“设置”，弹出“PLC 设定”对话框，在“内置

输入设置”选项卡中,最下方的选项“中断输入”的 IN1 选为“中断”,即将输入点 0.01 设置为中断输入点。

- ② 通过 CX-P 添加中断任务 141(输入中断 1)的新程序,编制中断处理用的程序。
- ③ 如图9.41 所示,通过 CX-P 编制周期执行任务 00 和中断任务 141。

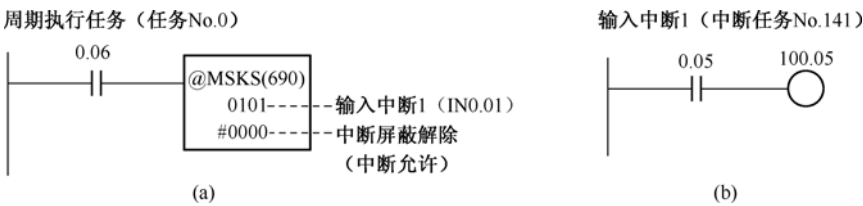


图 9.41 输入中断程序

- ④ 将工作模式切换到监控模式,按下列步骤执行程序,并观察结果。
- 闭合 0.06、0.05 后,观察闭合再断开 0.01 时,输出 100.05 的状态是否变化,是在上升沿还是下降沿发生变化。断开 0.05 后,观察闭合再断开 0.01 时,输出 100.05 的状态是否变化,是在上升沿还是下降沿发生变化。
- 闭合 0.06、0.01 后,观察闭合和断开 0.05 时,输出 100.05 的状态是否变化。
- 任务 00 再加一条 MSKS 指令操作数分别为 111 和#0001,按上两步的要求重做一遍。
- 将上一步中所加指令的第 2 个操作数改为#0000,再按第 1、2 步的要求重做一遍。

(3) 定时器中断控制指令使用练习

- ① 通过 CX-P 的 PLC 系统设定,将定时中断单位时间设定为 10 ms。
- ② 通过 CX-P 添加中断任务 02(间隔定时器 0)的新程序。
- ③ 如图9.42 所示,通过 CX-P 编制周期执行任务 00 和定时器中断任务 02。

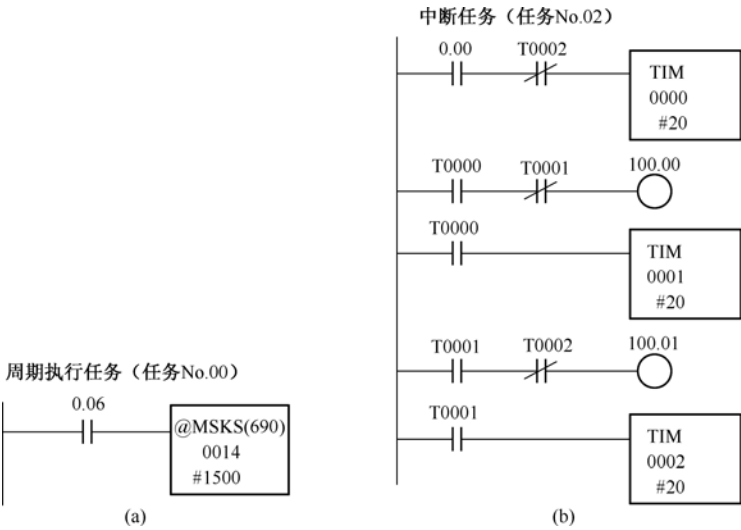


图 9.42 定时器中断程序

- ④ 将工作模式切换到监控模式,按下列步骤执行程序,并观察结果。
- 闭合开关 0.06 和 0.00,观察第一个 15 s、第 2 个 15 s,直至第 5 个 15 s 期间,TIM0000、TIM0001 和 TIM0002 的工作状态,以及输出 100.00 和 100.01 的状态。

- 当 100.00 和 100.01 刚转换为 ON 时，断开开关 0.00，100.00 和 100.01 的状态是否立即改变。
- 从上两步中总结出定时器中断的工作时序。

4. 实验报告内容

- ① 整理实验记录，分析图9.40参考程序的功能及子程序执行过程。
- ② 整理实验记录，分析图9.41 参考程序的功能及中断执行过程。
- ③ 整理实验记录，分析图9.42参考程序的功能及中断执行过程。

9.9 三人抢答器控制程序

1. 实验目的

- ① 设计 PLC 外部电路的接线图。
- ② 熟悉并采用符号编程方法，完成控制程序的输入。
- ③ 掌握程序调试的方法。

2. 控制要求

三人抢答器示意图如图9.43 所示。主持人配备“开始”和“复位”按钮各一个，抢答信号灯一个；三名参赛选手每人配备了“抢答”和“犯规”灯各一个，以及“抢答按钮”一个。

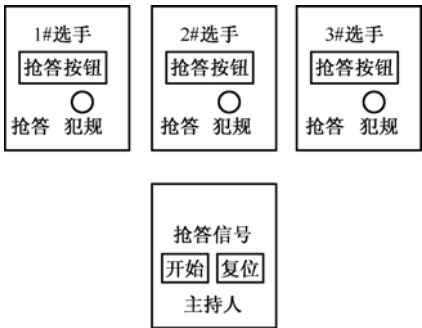


图 9.43 三人抢答器示意图

控制要求如下：

- ① 主持人通过控制台的按钮控制比赛的进行。主持人给出题目，并按下抢答“开始”按钮，此时抢答信号灯亮，提示各选手开始抢答。
- ② 当抢答结束后，主持人按下“复位”按钮，此时抢答信号灯灭，主持人开始准备出下一道抢答题。若抢答信号灯亮 10 s 内无人抢答，视为选手弃权，本题作废，同时抢答信号灯自动熄灭。
- ③ 在抢答信号灯亮后，先按下“抢答”按钮的选手，他面前的“抢答”灯常亮，后按的选手无效。答题完毕后，主持人按下“复位”按钮，使该选手的抢答灯熄灭。
- ④ 若主持人抢答信号灯未亮时，提前按下“抢答”按钮的选手被判犯规，他面前的“犯规”灯常亮(犯规扣分)，若有多名选手犯规，则只处罚第一个犯规的选手。主持人按下“复位”按钮，可将该选手的“犯规”灯熄灭。

3. I/O 分配

I/O 分配见表 9.2。

表 9.2 I/O 分配

输 入 点			输 出 点		
符 号	内 容	地 址	符 号	内 容	地 址
SB0	主持人开始按钮	0.00	L0	主持人抢答信号灯亮	100.00
SB1	1 号选手抢答按钮	0.01	L1	1 号选手抢答灯亮	100.01
SB2	2 号选手抢答按钮	0.02	L2	2 号选手抢答灯亮	100.02
SB3	3 号选手抢答按钮	0.03	L3	3 号选手抢答灯亮	100.03
SB4	主持人复位按钮	0.04	L4	1 号选手犯规灯亮	100.04
—	—	—	L5	2 号选手犯规灯亮	100.05
—	—	—	L6	3 号选手犯规灯亮	100.06



4. 程序设计

编程思路如下：

- ① 程序的总体结构可分为“抢答有效”与“抢答犯规”两部分，在此基础上分别分析抢答灯与犯规灯的输出条件。
  - ② 设计各选手的梯形图时，在逻辑条件中须串入其他选手抢答灯的常闭触点，旨在体现抢答器的基本功能——竞时锁定。在已有选手抢答之后其他选手再按自己的按钮将无效。犯规情况的编程与此相同。
  - ③ 设计主持人的程序时，须考虑正常抢答和 10 s 内无人抢答两种情况。
  - ④ 实现系统的总复位功能须在各逻辑行中串入复位按钮的常闭触点。
- 参考程序如图9.44所示。

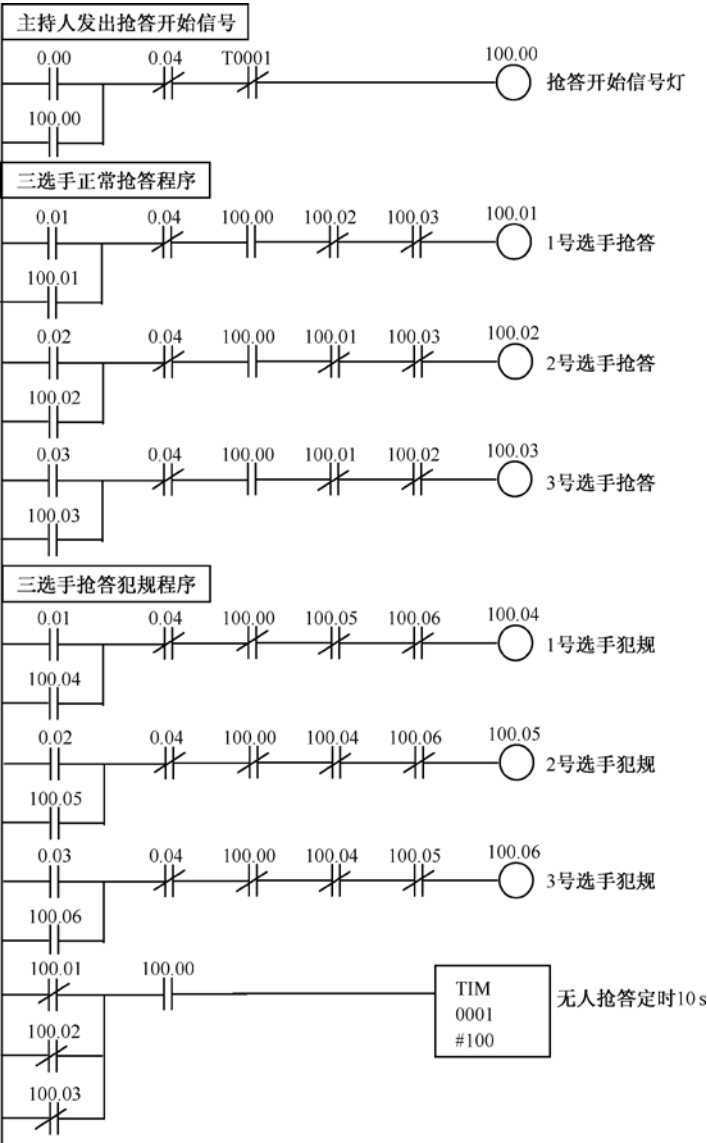


图 9.44 三人抢答器控制程序

5. 实验报告内容

- ① 画出 PLC 外部接线图、各按钮及指示灯的接线图。
- ② 采用符号编程的方法输入如图9.44所示的程序，并调试。
- ③ 总结调试的经验和教训。

9.10 变频器的基本运行

1. 实验目的

- ① 熟悉变频器面板操作和变频器参数初始化。
- ② 学会通过外部开关信号，控制变频器启停、正反转和频率。

2. 预习要求

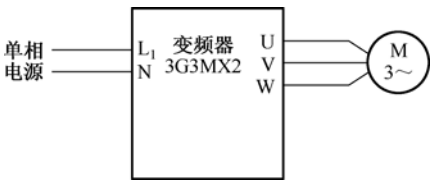
- ① 了解 3G3MX2 变频器的工作原理、组成、结构和连线。
- ② 熟悉变频器的面板操作和外部电路控制。

3. 实验内容和步骤

(1) 变频器的操作面板控制

用变频器操作面板完成变频器参数初始化，实现变频器(电动机)启停、频率调节和正反转。

- ① 根据 3G3MX2 的具体型号，选择单相电源或三相电源和电动机与之相连，如图9.45 所示。确认接线正确后，变频器加电。



- ② 参数初始化，设置频率设定值选择 A001=02，运行设定值选择 A002=02，输出频率设定 F001=5 Hz。表明选择变频器操作面板设定频率和控制变频器运行。

- ③ 设定运行方向选择 F004=00，即设为正转。
- ④ 通过参数查找，显示输出频率监控 d001。然后按回车键，此时显示的是 0.00 Hz。或通过按模式键 1 s 以上，就会显示 d001 的数据。

- ⑤ 按下运行键 RUN。此时运行指示灯变亮，同时电动机开始正转。d001 显示的数据由 0.00 Hz 逐渐增大到 5 Hz。

- ⑥ 在电动机运行的状态下，可以修改运行频率。找到参数 F001，设置 F001=8 Hz，立即观察到电动机转动加快。

- ⑦ 改变电动机的运行方向，须停止电动机。按下停止 STOP/RESET 键，电动机减速直至停止。找到参数 F004，设置 F004=01，即设电动机为反转。

- ⑧ 再按下运行键 RUN，则电动机反转。

(2) 变频器的外部开关控制

按如图9.46所示变频器的 2 线输入接线，确认接线正确后，接通变频器的电源。

- ① 参数初始化步骤如下。

- 设置频率设定值选择 A001=01，运行命令选择

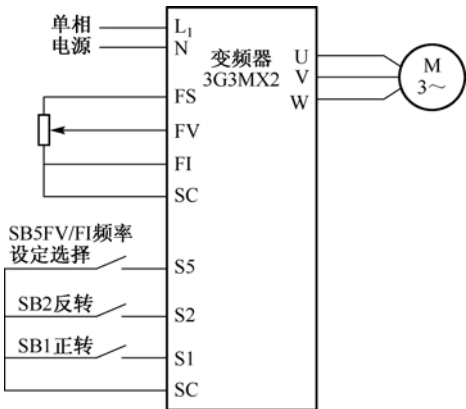


图 9.46 变频器的开关控制接线(2 线输入)

A002=01, 和电压/电流频率设定选择 A005=00。表明用变频器的外部接线端子控制其频率设定和运行。

- 用多功能输入 S1、S2 做变频器(电动机)的正反运行开关, 设置相应的 C001=00(正转)和 C002=01(反转)。
  - 选 S5 作为电压/电流频率设定选择, 须设置相应的 C005=16。令 S005=OFF, 为电压设定频率。
  - 显示输出频率监控(d001)然后按回车键。确定此时显示的是 0.00 Hz。
  - 采用调节外接电阻电位器模拟量电压设定频率, 通过 F001 观察使频率设定值为 10.00 Hz。设置电动机的加速时间 F002=25 s, 减速时间 F003=10 s。
- ② 按如图9.47所示正转开关 SB1 和反转开关 SB2 的动作时序, 操作步骤如下。
- 合上反转控制开关 SB2=ON。此时运行指示灯亮, 电动机逐渐加速到 15 Hz 的输出频率。
  - 停留一段时间 10 s 后, 断开反转控制开关 SB2=OFF, 电动机逐渐减速到输出频率 0 Hz。停止后运行指示灯会熄灭。

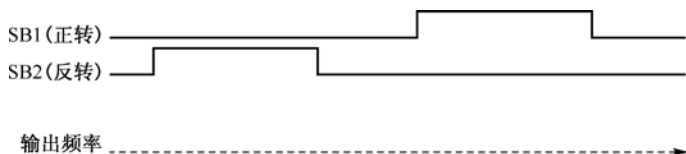


图 9.47 输出频率波形图

- 停留一段时间 6 s 后, 合上正转控制开关 SB1=ON。此时运行指示灯亮, 电动机按逐渐加速到输出频率 15 Hz。
- 停留一段时间例如 10 s 后, 断开正转控制开关 SB1=OFF, 电动机逐渐减速到 0 Hz。停止后运行指示灯会熄灭。
- 逐步增大模拟量电压设定的频率值(频率也可在运行命令之前设定好)。电动机的加速/减速时间通过 F002/F003(加速/减速时间)来设定。

完成上述步骤并画出图9.47的输出频率波形。

注意: 所设的加减速时间 F002/F003 是从 0 Hz 到最高频率的加/减速时间。

合上正转控制开关 SB1=ON, 待电动机加速到 15 Hz 的输出频率, 逐步增大模拟量(电压/电流)设定的频率值, 观察电动机的转速变化。再让 SB1=OFF, 电动机减速直至停止。

本例用电压进行频率设定时, 调节外接电阻电位器, 通过观察 F001 使频率设定为 15.00 Hz。

#### 4. 实验报告内容

- ① 整理实验记录, 总结变频器操作面板控制所需的参数初始化工作。
- ② 整理实验记录, 画出图9.47中的输出频率波形。
- ③ 采用计算的方法, 算出 15 Hz 的设定频率对应的模拟电压设定值。

## 9.11 变频器的多段速运行

### 1. 实验目的

- ① 掌握通过外部开关信号控制变频器的多段速运行。
- ② 掌握通过 PLC 的输出点控制变频器的多段速运行。

2. 预习要求

仔细阅读变频器的多段速控制内容。

3. 实验内容和步骤

要求输出频率曲线如图9.48所示。

(1) 通过外部开关信号控制变频器的多段速运行

采用开关控制变频器的输出频率，接线如图9.49所示。

控制开关设置：3G3MX2 的运行状态由 S1～S5(即开关 SB1～SB5)控制，S1 控制电动机正转，S2 控制电动机反转，S3、S4 为变频器的 3 段速控制输入端(3 段速只需 2 个输入端控制)，见表 9.3。S5 选用加减速时间选择端。

实验步骤如下。

① 参数设置。

在本例中选择 A019 = 00(用 4 端选择 16 段速)或 01(用 7 端选择 8 段速)都可，结果是一样的。3 个频率 0.00 Hz 、 50.00 Hz 和 25.00 Hz，对应电动机的 3 个转速，所以实际用 2 个控制端子 S3 和 S4 按表 9.3 选择，就可以得到 3 种转速。

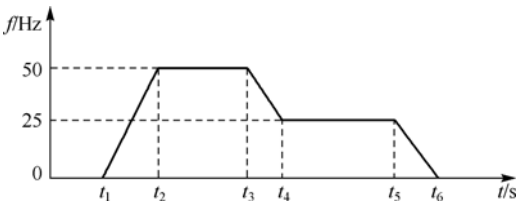


图 9.48 输出频率曲线

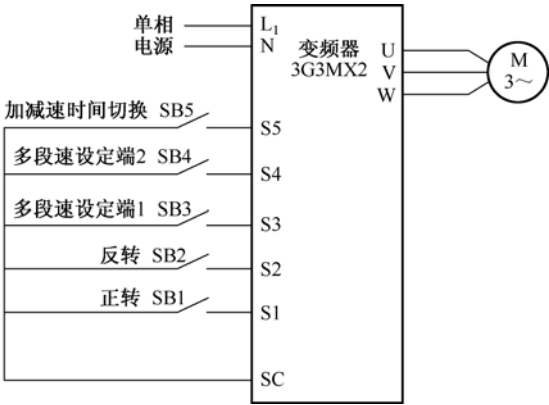


图 9.49 变频器接线

表 9.3 3 段速逻辑表

状 态 端 子	S4	S3	FOUT
0 段速	0	0	0 Hz
1 段速	0	1	50 Hz
2 段速	1	0	25 Hz

变频器的参数设置如下。

A004=50 Hz，最高频率。

A061=50 Hz，频率上限 50 Hz；A062=0.00 Hz，频率下限 0 Hz。

A019=00(用 4 端选择 16 段速)或 01(用 7 端选择 8 段速)。

A020=0 Hz，多段速 0 速。

A021=50 Hz，多段速 1 速。

A022=25 Hz，多段速 2 速。

F002=8 s, 加速时间 1 为 25 s, 即 0 Hz 升至 50 Hz 时间为 8 s。

F003=10 s, 减速时间 1 为 10 s, 即 50 Hz 降至 25 Hz 时间为 5 s。

A093=20 s, 减速时间 2 为 20 s, 即 25 Hz 降至 0 Hz 时间为 10 s。

A094=00, 通过 2 段加减速端子切换。

A001=01, A002=01, A005=00, 通过外部端子电路设置运行频率和运行命令。

C001=00(FW), S1 为正转控制端。

C002=01(RV), S2 为反转控制端。

C003=02(CF1), S3 为多段速设定端 1。

C004=03(CF2), S4 为多段速设定端 2。

C005=09(2CH), S5 为加减速时间切换端。

注意: 所设的加减速时间 F002/F003 和 A092/A093 是从 0 Hz 到最高频率的加/减速时间。这里设最高频率 A004=50 Hz, 则应设置 F002=8 s(为 0 Hz 升至 50 Hz 时间); 减速第一段 50 Hz 至 25 Hz 用 5 s, 则应设置 F003=10 s; 减速时第二段 50 Hz 至 10 Hz 为 10 s, 则应设置 A093=20 s。

#### ② 开关操作步骤。

- 第 1 阶段: 合上 SB1 和 SB3, 变频器用 8 s 由 0 Hz 用加速到 50 Hz, 并以 50 Hz 频率继续运行。
- 第 2 阶段: 同时断开 SB3 和合上 SB4, 变频器用 5 s 由 50 Hz 减速到 25 Hz 运行。
- 第 3 阶段: 停止阶段。断开 SB4 和合上 SB5, 从 25 Hz 减速到 0 Hz, 10 s 减速完成。

#### (2) 通过 PLC 的输出点控制变频器的多段速运行

PLC 控制变频器的多段速运行, 可以仿照(1)的各部分自行完成。内容有: 画出接线图, 完成输入端设置、参数设置、手动程序和自动程序的编写等。

### 4. 实验报告内容

- ① 在本实验中, 为什么 A019=00 或 01 都可以?
- ② 外部开关控制变频器多段速运行时, 画出对应图 9.48 输出频率曲线的开关动作次序。
- ③ PLC 控制变频器多段速运行时, 画出接线图, 完成输入端设置、参数设置、手动程序和自动程序的编写等。

## 参 考 文 献

- [1] 王冬青, 谭春. 欧姆龙 CP1H PLC 原理及应用[M]. 北京: 电子工业出版社, 2009.
- [2] 宫淑贞, 徐世许. 可编程序控制器原理及应用[M]. 2 版. 北京: 人民邮电出版社, 2009.
- [3] 徐世许, 宫淑贞, 彭涛. 可编程序控制器应用指南[M]. 北京: 电子工业出版社, 2007.
- [4] 谢克明, 夏路易. 可编程序控制器原理与程序设计[M]. 北京: 电子工业出版社, 2002.
- [5] 霍罡, 樊晓兵. 欧姆龙 CP1H PLC 应用基础与编程实践[M]. 北京: 机械工业出版社, 2008.
- [6] 祁文钊, 霍罡. CJ/CS 系列 PLC 应用基础及案例[M]. 北京: 机械工业出版社, 2006.
- [7] 杨公源, 黄琦兰. 可编程序控制器应用与实践[M]. 北京: 清华大学出版社, 2007.
- [8] 曹辉, 霍罡. 可编程序控制器过程控制技术[M]. 北京: 机械工业出版社, 2005.
- [9] 林育兹, 鲍平. 可编程序控制器原理及逻辑控制[M]. 北京: 机械工业出版社, 2005.
- [10] 林育兹, 谢炎基. 变频器应用案例[M]. 北京: 高等教育出版社, 2007.
- [11] 王立权, 王宗义, 王淑均, 徐伟. 可编程控制器原理与应用[M]. 哈尔滨: 哈尔滨工业大学出版社, 2005.
- [12] 吴亦锋. 可编程序控制器原理与应用速成[M]. 福州: 福建科学技术出版社, 2005.
- [13] OMRON. CP1H CPU 单元操作手册, 2007.
- [14] OMRON. CP1L CPU Unit Operation Manual, 2009.
- [15] OMRON. CP1 系列编程手册, 2007.
- [16] OMRON. CP1E CPU Unit Hardware User's Manual, 2010.
- [17] OMRON. CP1E CPU 单元指令参考手册, 2010.
- [18] OMRON. WS02-CXPC-V9 CX-Programmer 9 Operation Manual, 2009.
- [19] OMRON. 3G3MX2 多功能小型变频器用户手册, 2010.
- [20] OMRON. CJ1M CPU 单元操作手册, 2002.
- [21] OMRON. CJ1M CPU 单元编程手册, 2002.
- [22] OMRON. CJ1M CPU 单元编程手册, 2002.
- [23] <http://www.fa.omron.com.cn>.